

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ  
НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
“ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ”

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

к лабораторным работам

**«Программирование и решение дифференциальных уравнений  
в системе MathCAD»**

из раздела «Использование программ в среде Windows»

дисциплины «Основы программирования  
и информационных технологий»

для студентов направления подготовки

6.050801 «Микро- и нанoeлектроника»

**Методические указания** к лабораторным работам «Программирование и решение дифференциальных уравнений в системе MathCAD» из раздела «Использование программ в среде Windows» дисциплины «Основы программирования и информационных технологий» для студентов направления подготовки 6.050801 «Микро- и нанoeлектроника» / Состав.: Р.В. Зайцев, Е.С. Клепикова, А.В. Момотенко. – Харьков: НТУ «ХПИ», 2014. – 59 с.

Составители: Р.В. Зайцев  
Е.С. Клепикова  
А.В. Момотенко

Рецензент доц. И.В. Федорин

Кафедра физического материаловедения для электроники и гелио-энергетики



## ВВЕДЕНИЕ

Методические указания к лабораторным работам по разделу «Использование программ в среде Windows» дисциплины «Основы программирования и информационных технологий» касаются двух лабораторных работ: «Программирование и регрессия в MathCAD» и «Решение дифференциальных уравнений и систем».

Универсальная математическая система MathCAD является одной из лучших систем для научно-технических вычислений. В среде Mathcad доступны более сотни операторов и логических функций, предназначенных для численного и символьного решения технических проблем различной сложности. Она имеет мощные средства для реализации численных методов расчета, возможность выполнения многих операций символьной математики. Исходные данные и результаты вычислений представляются в естественном математическом виде.

Mathcad содержит:

- обширную библиотеку встроенных математических функций;
- инструменты построения графиков различных типов;
- средства создания текстовых комментариев и оформления отчетов;
- конструкции, подобные программным конструкциям языков программирования, позволяющие писать программы для решения задач, которые невозможно или очень сложно решить стандартными инструментами пакета;
- удобно организованную интерактивную систему получения справки и оперативной подсказки.

Программные средства такого рода называют универсальными математическими пакетами, системами или средами. Как в электронных таблицах, любое изменение содержимого рабочего документа MathCAD вызывает обновление всех зависимых результатов и перерисовку графиков. Объединяя в одном рабочем листе текст, графику и математические выкладки, Mathcad облегчает понимание самых сложных технических вычислений.

## ЛАБОРАТОРНАЯ РОБОТА 1

### ПРОГРАММИРОВАНИЕ И РЕГРЕССИЯ В MathCAD

Цель работы – Научиться работать с программой MathCAD. Научиться проводить математические вычисления и представлять полученные результаты в графической форме с помощью программы MathCAD. Научиться составлять программы средствами MathCAD для выполнения вычислений, не предусмотренных встроенными функциями.

#### 1.1 Общие сведения

MathCAD имеет не очень мощный, но весьма элегантный собственный язык. С одной стороны, он дает возможность программисту эффективно применять программный код в документах MathCAD. С другой, простота и интуитивность языка программирования позволяет быстро ему обучиться. Наконец, программные модули внутри документа MathCAD сочетают в себе и обособленность (поэтому их легко отличить от остальных формул), и простоту смыслового восприятия. Несмотря на небольшое число операторов, язык программирования MathCAD позволяет решать самые различные, в том числе и довольно сложные, задачи и является серьезным подспорьем для расчетов.

Отметим, что возможность составлять программы реализована только в версии Professional.

##### 1.1.1 Программирование на MathCAD «без программирования»

В ранних версиях MathCAD встроенного языка программирования не было. Чтобы применять привычные операции проверки условий и организовывать циклы, приходилось изобретать причудливую смесь из встроенных функций условия **if** (листинг 1.1) и **until** и комбинаций ранжированных переменных (листинг 1.2).

##### Примечание

*В связи с устоявшимися традициями применения языка программирования функцию **until** настоятельно не рекомендуется использовать в дальнейшей работе (тем не менее, она действует в MathCAD, но отнесена к устаревшим функциям).*

Листинг 1.1. Функция условия

```
f(x) := if (x < 0, "negative", "positive")  
f(1) = "positive"  
f(-1) = "negative"
```

Листинг 1.2. Организация цикла при помощи ранжированной переменной

```
i := 0 .. 10  
xi := i2
```

Фактически, использование ранжированных переменных – мощный аппарат MathCAD, похожий на применение циклов в программировании. В подавляющем большинстве случаев намного удобнее организовать циклы (в том числе вложенные) с помощью ранжированных переменных, чем заниматься для этого программированием. Полезнее освоить технику, связанную с ранжированными переменными, векторами и матрицами, поскольку на ней основаны главные принципы расчетов в MathCAD, в частности подготовка графиков.

### 1.1.2 Язык программирования MathCAD

Для вставки программного кода в документы в MathCAD имеется специальная панель инструментов «**Programming**» (**Программирование**), которую можно вызвать на экран нажатием кнопки «**Programming Toolbar**» на панели «**Math**» (**Математика**), как показано на рис. 1.1. Большинство кнопок этой панели выполнено в виде текстового представления операторов программирования, поэтому их смысл легко понятен. Изложим последовательно основные составные части языка программирования MathCAD и рассмотрим примеры его использования.

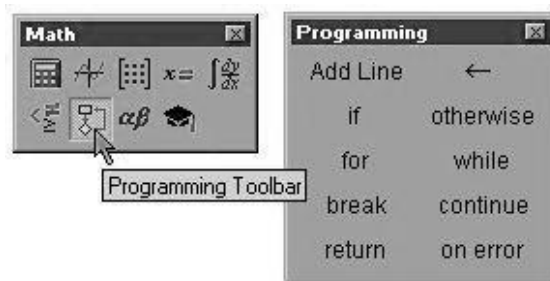


Рисунок 1.1 - Панель инструментов «**Programming**»

#### 1.1.2.1 Что такое программа?

Основными инструментами работы в MathCAD являются математические выражения, переменные и функции. Нередко записать формулу, использующую ту или иную внутреннюю логику (например, возвращение различных значений в зависимости от условий), в одну строку не удастся. Назначение программных модулей как раз и заключается в определении выражений, переменных и функций в несколько строк, часто с приме-

нием специфических программных операторов. Сравните определение функции  $f(x)$  из листинга 1.1 с определением  $f(x)$  с помощью программного модуля (листинг 1.3).

*Листинг 3.3. Функция условия, определенная с помощью программы*

```
f{x} := | "negative"  if x < 0  
        | "positive"  if x > 0  
        | "zero"     otherwise  
f{1} = "positive"  
f{-1} = "negative"  
f{0} = "zero"
```

Несмотря на принципиальную эквивалентность определения функций и переменных через встроенные функции MathCAD или программные модули, программирование имеет ряд существенных преимуществ, которые в ряде случаев делают документ более простым и читаемым:

- возможность применения циклов и условных операторов;
- простота создания функций и переменных, требующих нескольких простых шагов (как в примере листинга 1.3);
- возможность создания функций, содержащих закрытый для остального документа код, включая преимущества использования локальных переменных и обработку исключительных ситуаций (ошибок).

Как видно из листинга 1.3, программный модуль обозначается в MathCAD вертикальной чертой, справа от которой последовательно записываются операторы языка программирования.

#### 1.1.2.2 Создание программы (Add Line)

Чтобы создать программный модуль, например, представленный в предыдущем разделе (см. листинг 1.3):

1. Введите часть выражения, которая будет находиться слева от знака присваивания и сам знак присваивания. В нашем примере это имя функции  $f(x)$ .

2. При необходимости вызовите на экран панель инструментов «**Programming**» (**Программирование**) (см. рис. 1.1).

3. Нажмите на этой панели кнопку «**Add Line**» (**Добавить линию**).

4. Если приблизительно известно, сколько строк кода будет содержать программа, можно создать нужное количество линий повторным нажатием кнопки «**Add Line**» (**Добавить линию**) соответствующее число раз (на рис. 1.2 показан результат трехкратного нажатия).

5. В появившиеся местозаполнители введите желаемый программный код, используя программные операторы.

В рассматриваемом примере в каждый местозаполнитель вводится строка, например, "positive" (рис. 1.3), затем нажимается кнопка **If** (Если) на панели «**Programming**» (**Программирование**) и в возникший местозаполнитель вводится выражение  $x > 0$  (рис. 1.4).

После того как программный модуль полностью определен, и ни один местозаполнитель не остался пустым, функция может использоваться обычным образом, как в численных, так и в символьных расчетах.

Не вводите с клавиатуры имена программных операторов. Для их вставки можно применять лишь сочетания клавиш, которые приведены в тексте всплывающей подсказки (рис. 1.2 и 1.3).

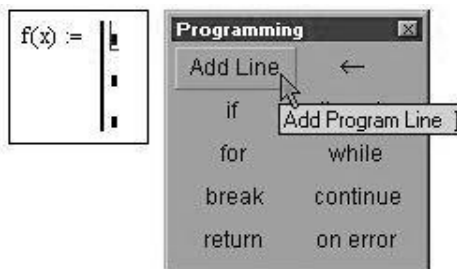


Рисунок 1.2 - Начало создания программного модуля

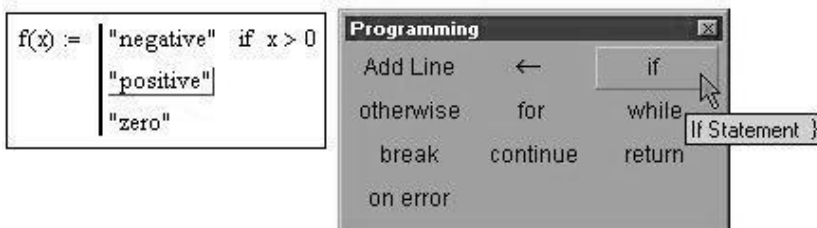


Рисунок 1.3 - Вставка программного оператора

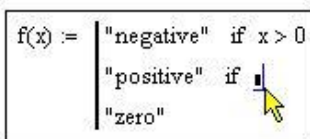


Рисунок 1.4 - Вставка условия в программу

#### 1.1.2.3 Разработка программы

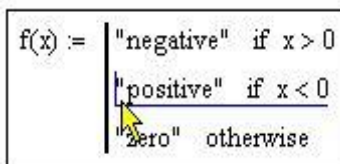
Вставить строку программного кода в уже созданную программу можно в любой момент с помощью той же самой кнопки **Add Line** (Добавить линию). Для этого следует предварительно поместить на нужное место внутри программного модуля линии ввода. Например, расположение



линии ввода на строке, показанной на рис. 1.5, приведет к появлению новой линии с место-заполнителем перед этой строкой. Если передвинуть вертикальную линию ввода из начала строки (как на рис. 1.5) в ее конец, то новая линия появится после строки. Если выделить строку не целиком, а лишь некоторую ее часть (рис. 1.6), то это повлияет на положение в программе новой строки кода (результат нажатия кнопки **Add Line** показан на рис. 1.7).

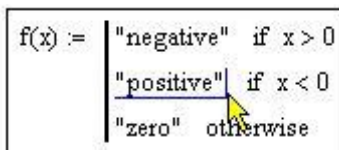
#### Совет

*Не забывайте, что для желаемого размещения линий ввода внутри формулы можно использовать не только мышь и клавиши со стрелками, но и пробел. С помощью последовательных нажатий пробела линии ввода "захватывают" разные части формулы.*



```
f(x) := | "negative" if x > 0  
        | "positive" if x < 0  
        | "zero" otherwise
```

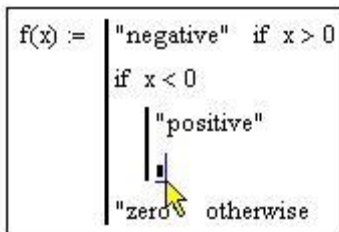
Рисунок 1.5 - Вставка новой строки в существующую программу



```
f(x) := | "negative" if x > 0  
        | "positive" if x < 0  
        | "zero" otherwise
```

Рисунок 1.6 - Положение линий ввода влияет на положение новой линии

Зачем может потребоваться вставка новой линии в положение, показанное на рис. 1.7? Новая вертикальная черта с двумя линиями выделяет фрагмент программы, который относится к условию  $x > 0$ , находящемуся в его заголовке. Пример возможного дальнейшего программирования показан в листинге 1.4.



```
f(x) := | "negative" if x > 0  
        | if x < 0  
          | "positive"  
          |  
        | "zero" otherwise
```

Рисунок 1.7 - Результат вставки новой линии в программу

Листинг 1.4. Пример усовершенствования программы

```
f(x) := | "negative"  if x < 0
        |
        | if x > 0
        | | "positive"
        | | "big positive"  if x > 1000
        | "zero"  otherwise
f(1) = "positive"
f(105) = "big positive"
```

В режиме выполнения программы, а это происходит при любой попытке вычислить  $f(x)$ , выполняется последовательно каждая строка кода. Например, в предпоследней строке листинга 3.4 вычисляется  $f(i)$ . Рассмотрим работу каждой строки кода этого листинга.

1. Поскольку  $x=1$ , то условие  $x < 0$  не выполнено, и в первой строке ничего не происходит.
2. Условие второй строки  $x > 0$  выполнено, поэтому выполняются обе следующие строки, объединенные короткой вертикальной чертой в общий фрагмент.
3. Функции  $f(x)$  присваивается значение  $f(x) = \text{"positive"}$ .
4. Условие  $x > 1000$  не выполнено, поэтому значение "big positive" не присваивается  $f(x)$ , она так и остается равной строке "positive".
5. Последняя строка не выполняется, т. к. одно из условий ( $x > 0$ ) оказалось истинным, и оператор **otherwise** (т. е., "иначе") не понадобился.

Таким образом, основной принцип создания программных модулей заключается в правильном расположении строк кода. Ориентироваться в их действии довольно легко, т. к. фрагменты кода одного уровня сгруппированы в программе с помощью вертикальных черт.

#### 1.1.2.4 Локальное присваивание (<-)

Язык программирования MathCAD не был бы эффективным, если бы не позволял создавать внутри программных модулей локальные переменные, которые "не видны" извне, из других частей документа. Присваивание в пределах программ, в отличие от документов MathCAD, производится с помощью оператора **Local Definition** (Локальное присваивание), который вставляется нажатием кнопки с изображением стрелки <- на панели **Programming** (Программирование).

Ни оператор присваивания  $:=$ , ни оператор вывода  $=$  в пределах программ не применяются.

Локальное присваивание иллюстрируется листингом 1.5. Переменная  $z$  существует только внутри программы, выделенной вертикальной чертой. Из других мест документа получить ее значение невозможно.

### Листинг 3.5. Локальное присваивание в программе

```
f(x) :=  $\begin{cases} z \leftarrow 4 \\ z + x \end{cases}$   
f(1) = 5
```

#### 1.1.2.5 Условные операторы (if, otherwise)

Действие условного оператора **if** состоит из двух частей. На рис. 1.8 показан пример вставки условного оператора. Сначала проверяется логическое выражение (условие) справа от него. Если оно истинно, выполняется выражение слева от оператора **if**. Если ложно – ничего не происходит, а выполнение программы продолжается переходом к ее следующей строке.

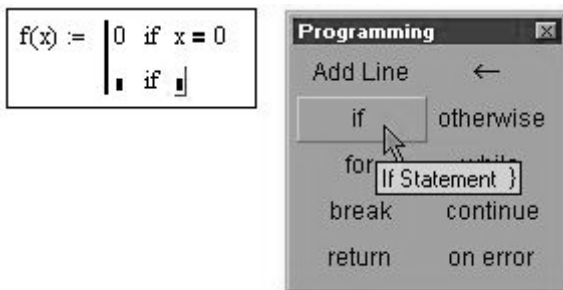


Рисунок 1.8 - Вставка условного оператора

Вставить условный оператор в программу можно следующим образом:

1. Если необходимо, введите левую часть выражения и оператор присваивания.
2. Создайте новую строку программного кода, нажав на панели **Programming** (Программирование) кнопку **Add Line** (Добавить строку).
3. Нажмите кнопку условного оператора **if**.
4. Справа от оператора **if** введите условие. Пользуйтесь логическими операторами, вводя их с панели **Boolean** (Булевы операторы).
5. Выражение, которое должно выполняться, если условие оказывается выполненным, введите слева от оператора **if**.
6. Если в программе предусматриваются дополнительные условия, добавьте в программу еще одну строку нажатием кнопки **Add Line** и введите их таким же образом, используя оператор **if** или **otherwise**. Оператор **otherwise** используется совместно с одним или несколькими условными операторами **if** и указывает на выражение, которое будет выполняться, если ни одно из условий не оказалось истинным.

Примеры использования операторов **if** и **otherwise** приведены ранее

(см. листинги 1.3 и 1.4).

#### 1.1.2.6 Условные операторы (if, otherwise)

В языке программирования MathCAD имеются два оператора цикла: **for** и **while**. Первый из них дает возможность организовать цикл по некоторой переменной, заставляя ее пробегать некоторый диапазон значений. Второй создает цикл с выходом из него по некоторому логическому условию. Чтобы вставить в программный модуль оператор цикла:

1. Создайте в программном модуле новую линию.
2. Вставьте один из операторов цикла **for** или **while** нажатием одноименной кнопки на панели **Programming** (Программирование).
3. Если выбран оператор **for** (рис. 1.9), то вставьте в соответствующие местозаполнители имя переменной и диапазон ее значений (листинги 1.6 и 1.7), а если **while** - то логическое выражение, при нарушении которого должен осуществляться выход из цикла (листинг 1.8).

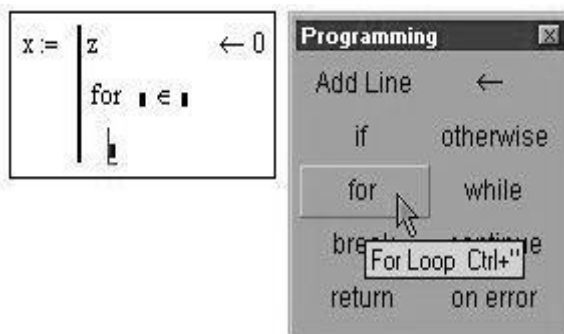


Рисунок 1.9 - Вставка оператора цикла

Листинг 1.6. Оператор цикла **for** с ранжированной переменной

```
x := | z ← 0
      | for i ∈ 0.. 5
      |   z ← z + i
x = 15
```

Листинг 1.7. Оператор цикла **for** с вектором

```
x := | z ← 0
      | for i ∈ {1 2 3}
      |   z ← z + i
x = 6
```

Листинг 1.8. Оператор цикла **while**,

```

x := | z ← 0
      | while z < 10
      |   z ← z + 1
x = 10

```

4. В нижний местозаполнитель введите тело цикла, т. е. выражения, которые должны выполняться циклически.

5. При необходимости дополните программу другими строками и введите в них нужный код.

#### Примечание

Диапазон значений переменной в условии цикла **for** можно задать как с помощью диапазона ранжированной переменной (листинг 1.6), так и с помощью вектора (листинг 1.7).

Иногда необходимо досрочно завершить цикл, т. е. не по условию в его заголовке, а в некоторой строке в теле цикла. Для этого предназначен оператор **break**. Модификации листингов 1.6 и 1.8 с прерыванием цикла оператором **break** приведены в листингах 1.9 и 1.10, соответственно. Например, в листинге 1.9, как только значение переменной цикла *i* достигает 2, цикл, благодаря оператору **break** в последней строке программного модуля, прерывается. Соответственно, значение переменной *x* остается равным  $0+1+2=3$ .

Листинг 1.9. Оператор **break** внутри цикла **for**

```

x := | z ← 0
      | for i ∈ 0.. 5
      |   | z ← z + i
      |   | break if i = 2
x = 3

```

Листинг 1.10. Оператор **break** внутри цикла **while**

```

x := | z ← 0
      | while z < 10
      |   | z ← z + 1
      |   | break if z > 5
x = 6

```

#### Примечание

Чтобы четче обозначить границы завершения тела цикла, в его конце может использоваться дополнительная строка с оператором **continue**, который вводится одноименной кнопкой панели **Programming**. Примеры, модернизирующие листинги 1.7 и 1.8, иллюстрируются ли-

стингами 1.11 и 1.12, соответственно. Как видно, на результат программы наличие оператора **continue** не влияет.

Листинг 1.11. Оператор **continue** в конце цикла **while**

```
x := | z ← 0
      | while z < 10
      |   | z ← z + 1
      |   | continue
x = 10
```

Листинг 1.12 Оператор **continue** в конце цикла **for**

```
x := | z ← 0
      | for i ∈ {1 2 3}
      |   | z ← z + i
      |   | continue
x = 6
```

#### 1.1.2.7 Возврат значения (return)

Если для определения переменной или функции применяется программный модуль, то его строки исполняются последовательно при вычислении в документе этой переменной или функции. Соответственно, по мере выполнения программы рассчитываемый результат претерпевает изменения. В качестве окончательного результата выдается последнее присвоенное значение (примеры можно найти в листингах 1.3 – 1.12). Чтобы подчеркнуть возврат программным модулем определенного значения, можно взять за правило делать это в последней строке программного модуля (листинг 1.13).

Листинг 1.13. Возврат значения обозначен явно в последней строке программы

```
f(x) := | y ← x2
        | z ← y + 1
        | z
f(2) = 5
```

Вместе с тем, можно прервать выполнение программы в любой ее точке (например, с помощью условного оператора) и выдать некоторое значение, применив оператор **return**. В этом случае при выполнении указанного условия (листинг 1.14) значение, введенное в местозаполнитель

после **return**, возвращается в качестве результата, а никакой другой код больше не выполняется. Вставляется в программу оператор **return** с помощью одноименной кнопки панели **Programming** (Программирование).

Листинг 1.14. Применение оператора **return**

```
f(x) :=  $\left\{ \begin{array}{l} z \leftarrow x^2 \\ \text{return } "zero" \text{ if } x = 0 \\ \text{return } "i" \text{ if } x = i \\ z \end{array} \right.$ 

f(-1) = 1
f(2) = 4
f(0) = "zero"
f(i) = "i"
```

#### 1.1.2.8 Перехват ошибок (on error)

Программирование в MathCAD позволяет осуществлять дополнительную обработку ошибок. Если пользователь предполагает, что выполнение кода в каком-либо месте программного модуля способно вызвать ошибку (например, деление на ноль), то эту ошибку можно перехватить с помощью оператора **on error**. Чтобы вставить его в программу, надо поместить линии ввода в ней в нужное положение и нажать кнопку с именем оператора **on error** на панели **Programming** (Программирование). В результате появится строка с двумя местозаполнителями и оператором **on error** посередине (рис. 1.10).

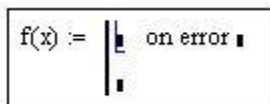


Рисунок 1.10 - Вставка оператора перехода по ошибке

В правом местозаполнителе следует ввести выражение, которое должно выполняться в данной строке программы. В левом - выражение, которое будет выполнено вместо правого выражения, если при выполнении последнего возникнет ошибка. Приведем пример применения оператора **on error** (листинг 1.15) в программном модуле, который рассчитывает функцию обратного числа значению **n**. Если **n**  $\neq 0$ , то и присвоенное значение **z**  $\neq 0$ , поэтому в последней строке программы выполняется правое выражение расчета **1 / z**. Так происходит при расчете **f(-2)**. Если попытаться вычислить **f(0)**, как в конце листинга, то выполнение программы, заложенной в **f(n)**, вызовет ошибку деления на ноль в последней строке программы. Соответственно, вместо выражения справа от оператора **on**

**error** будет выполнено левое выражение, присваивающее функции, **f(n)** строковое значение "user error: cannot divide by zero" (пользовательская ошибка: деление на ноль невозможно). Конечно, этой строке можно присвоить и текст на русском языке.

Листинг 1.15. Перехват ошибки деления на ноль

```
f(n) := 
$$\left| \begin{array}{l} z \leftarrow n \\ \\ \text{"user error: can't divide by zero"} \quad \text{on error} \quad \frac{1}{z} \end{array} \right.$$

```

$f(-2) \rightarrow \frac{-1}{2}$

$f(0) = \text{"user error: can't divide by zero"}$

Оператор перехвата ошибок удобно применять в комбинации со встроенной функцией **error(S)**. Она приводит к генерации ошибки в обычной для MathCAD форме с сообщением s. Пример усовершенствования листинга 1.15 для такого стиля обработки ошибки деления на ноль показан на рис. 1.11. Обратите внимание, что сделанные изменения свелись к помещению текста сообщения об ошибке в аргумент функции **error**.

```
f(n) := 
$$\left| \begin{array}{l} z \leftarrow n \\ \\ \text{error("user error: can't divide by zero")} \quad \text{on error} \quad \frac{1}{z} \end{array} \right.$$

```

$f(-2) \rightarrow \frac{-1}{2}$

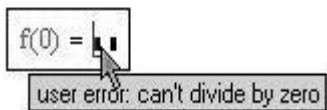


Рисунок 1.11 - Перехват ошибки деления на ноль

Ниже в таблице 1.1 приведены операторы, находящиеся на кнопках панели программирования.



Таблица 1.1 – Операторы, находящиеся на кнопках панели программирования

Команда	Функция	Пример
<b>Add Line</b>	Добавляет новую строку под/над (зависит от выделения) текущей строкой.	
$\leftarrow$	Присваивание значения локальной переменной.	$y \leftarrow 0$
<b>if</b>	Условный оператор (оператор ветвления) <b>if</b> ; условие должно стоять после <b>if</b> , а оператор, который выполняется, если выполнено заданное условие, – перед <b>if</b> .	$f(x) := \begin{cases} -x & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$ $f(-3) = 3 \qquad f(4) = 4$
<b>other-wise</b>	Обозначает оператор, который должен быть исполнен, если условие оператора <b>if</b> не выполняется.	$f(x) := \begin{cases} -x & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$ $f(-3) = 3 \qquad f(4) = 4$
<b>for</b>	Цикл <b>for</b> ; за ключевым словом <b>for</b> следует переменная-счетчик, а после символа принадлежности вводится промежуток изменения этой переменной. Внутренние операторы цикла сдвинуты немного вправо.	$\text{Sum}(n) := \begin{cases} s \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad s \leftarrow s + i \end{cases}$ $\text{Sum}(4) = 10$
<b>while</b>	Цикл <b>while</b> ;внутренние операторы цикла будут исполняться до тех пор, пока будет истинным условие, следующее за ключевым словом <b>while</b> . Пример показывает применение цикла для нахождения нулей функции методом касательных Ньютона.	$N(x, f, f_x) := \text{while }  f(x)  > \epsilon$ $\qquad x \leftarrow x - \frac{f(x)}{f_x(x)}$ $N(2, \sin, \cos) = 3.142$
<b>break</b>	Служит для преждевременного завершения цикла, чтобы, например, избежать закливания или слишком продолжительных вычислений.	$\text{break if } i \geq 10$
<b>continue</b>	Служит для преждевременного завершения текущей итерации цикла; сам цикл при этом продолжается.	$\text{continue if } x \geq 10$
<b>return</b>	Преждевременное завершение программы; указанное в ячейке значение будет возвращено.	$\text{return } y$
<b>on error</b>	Если при вычислении выражения $\text{expr2}$ возникла ошибка, вычисляется выражение $\text{expr1}$ .	$\text{expr1 on error expr2}$

### 1.1.3 Примеры программирования

Рассмотрим два простых примера использования программных модулей в MathCAD для численных (листинг 1.16) и символьных (листинг 1.17) расчетов. В двух приведенных листингах используется большинство операторов, рассмотренных в данной главе. Когда вы станете сами разрабатывать свои программные модули в MathCAD, не забывайте, что операторы программирования вставляются в текст программы с помощью кнопок панели инструментов **Programming (Программирование)**. Их имена нельзя ни в коем случае просто набивать на клавиатуре, поскольку они не будут восприняты MathCAD корректно.

Листинг 1.16. Программирование в численных расчетах

```
f(n) := | return -99 if n < 0
        | z ← 1
        | for i ∈ 1.. n
        |   z ← z · i
        | z
f(-2) → -99
f(0) = 0
f(3.9) = 6
f(3) = 6
f(10) = 3.629 × 106
```

Листинг 1.17. Программирование в символьных расчетах

```
f(n) := | -1 if n < 0
        | x on error  $\frac{d^n}{dx^n} x^{10}$  otherwise
f(1) → 10 · x9
f(10) → 3628800
f(-3) → -1
f(2.1) → x
```

Рассмотрим пример программного блока, вычисляющего факториал с использованием рекурсии. В данном примере (листинг 1.18) возвращается, только если  $n=0$  или  $n=1$ . Обратите внимание на задание несколь-

ких условий - со знаком плюс, каждое условие в скобках. В остальных случаях, учитывая формулу  $n! = n \cdot (n-1)!$ , вызывается  $fakt(n-1)$  и умножается на  $n$ . При помощи функции **error** можно вывести сообщение об ошибке при неправильном вводе аргумента.

Листинг 1.18. Вычисление факториала с использованием рекурсии

```
fakt(n) := | error("must be positive!") if n < 0
           | error("must be integer!") if n ≠ floor(n)
           | 1 if [(n = 0) + (n = 1)]
           | (n·fakt(n - 1)) otherwise
```

fakt(6) = 720    6! = 720

fakt(-4) = ■ ■

must be positive!

fakt(3.4) = ■ ■

must be integer!

Для демонстрации работы цикла **for** рассмотрим ту же задачу, но вычисление факториала осуществим при помощи цикла, как показано на листинге 1.19. Если  $n = 0$  или  $n = 1$ , возвращается 1, в противном случае при помощи цикла **for** вычисляется произведение  $n! = 1 * 2 * 3... * n$ . Вычисленное последним значение **p** возвращается автоматически.

Листинг 1.19. Вычисление факториала с использованием цикла

```
fakt1(n) := | 1 if [(n = 0) + (n = 1)]
           | otherwise
           | | p ← 1
           | | for i ∈ 1..n
           | | p ← p·i
```

fakt1(6) = 720

В следующем примере при помощи алгоритма Евклида определяется наибольший общий делитель. Для реализации алгоритма используется цикл с ключевым словом **while**, как показано на листинге 1.20.

Листинг 1.20. Вычисление наибольшего общего делителя с использованием алгоритма Евклида

$$\text{ggf}(x, y) := \begin{array}{|l} x \text{ if } y = 0 \\ \text{while } y \neq 0 \\ \quad \left| \begin{array}{l} h \leftarrow y \\ y \leftarrow \text{mod}(x, y) \\ x \leftarrow h \end{array} \right. \end{array} \quad \begin{array}{l} \text{ggf}(25, 35) = 5 \\ \text{ggf}(8, 0) = 8 \\ \text{ggf}(8, 24) = 8 \end{array}$$

В следующих листингах рассматривается прерывание цикла при помощи операторов **break** и **continue**, а также оператор **return**. Листинг 1.21 – реализация метода касательных Ньютона для определения нулей функции. На основе начального значения  $x$  вычисляется новое улучшенное значение  $x$ , расположенное ближе к искомому нулю функции. При этом итерации повторяются до тех пор, пока значение функции не станет меньше заданной точности (в листинге –  $10^{-6}$ ). При помощи оператора **return** организовано завершение программы в нужный момент. В данном примере если число итераций больше или равно 10, то происходит прерывание программы и выдается сообщение о том, что слишком много итераций. Здесь также отслеживаются случаи, когда производная в знаменателе близка к нулю и выдается об этом сообщение.

Листинг 1.21. Реализация метода касательных Ньютона для определения нулей функции.

$$\text{Newton}(x, \text{ff}, \text{ff}_x) := \begin{array}{|l} i \leftarrow 0 \\ \text{while } |\text{ff}(x)| > 10^{-6} \\ \quad \left| \begin{array}{l} i \leftarrow i + 1 \\ \text{break if } i \geq 10 \\ \text{return "Derivative is 0!" if } |\text{ff}_x(x)| < 10^{-6} \\ x \leftarrow x - \frac{\text{ff}(x)}{\text{ff}_x(x)} \end{array} \right. \\ \text{return "Too mach iterations!" if } i \geq 10 \\ \text{return } x \text{ otherwise} \end{array}$$

$$\text{ff}(x) := \sin(x) - 1 + x \quad \text{ff}_x(x) := \frac{d}{dx} \text{ff}(x) \quad x := 1$$

$$X := \text{Newton}(x, \text{ff}, \text{ff}_x) \quad X = 0.511 \quad \text{root}(\text{ff}(x), x) = 0.511$$

$$x := \pi \quad X := \text{Newton}(x, \text{ff}, \text{ff}_x) \quad X = \text{"Derivative is 0!"}$$

$$x := -100 \quad X := \text{Newton}(x, \text{ff}, \text{ff}_x) \quad X = \text{"Too mach iterations!"}$$

Кроме ключевого слова **break** имеется ключевое слово **continue** с похожей функцией. В то время как **break** прерывает цикл и осуществляет переход к следующему за циклом оператору, **continue** прерывает выполнение только текущей итерации.

В примере на листинге 1.22 ключевое слово **continue** используется для выявления всех нулей функции на заданном интервале. При этом в примере производится разбиение интервала на  $n$  равных подинтервалов, и ищутся те из них, на которых функция меняет знак. При обнаружении такого подинтервала вызывается функция, реализующая метод касательных Ньютона, с начальным значением, находящимся в середине подинтервала.

*Листинг 1.22. Реализация метода касательных Ньютона для определения нулей функции.*

$$\begin{aligned}
 ff(x) &:= \left(x - \frac{1}{2}\right) \cdot (x+1) \cdot (x-2) & ff'_x(x) &:= \frac{d}{dx} ff(x) \\
 null(ff, ff'_x, a, b, n) &:= \left| \begin{array}{l}
 ans \leftarrow 0 \\
 \text{for } i \in 0..n-1 \\
 \quad x_{left} \leftarrow a + \frac{(b-a)}{n} \cdot i \\
 \quad x_{right} \leftarrow a + \frac{(b-a)}{n} \cdot (i+1) \\
 \quad \text{continue if } ff(x_{left}) \cdot ff(x_{right}) > 0 \\
 \quad x \leftarrow \frac{x_{left} + x_{right}}{2} \\
 \quad x_{nw} \leftarrow \text{Newton}(x, ff, ff'_x) \\
 \quad ans \leftarrow ans + 1 \\
 \end{array} \right| x_{nw} \\
 null(ff, ff'_x, -1, 3, 15) &= \begin{pmatrix} -1 \\ 0.5 \\ 2 \end{pmatrix}
 \end{aligned}$$

Обратите внимание на то, что по завершении работы программы выдается вектор значений.

На листинге 1.23 приводится программа, вычисляющая коэффициенты Фурье функции, причем в результате выдается матрица значений: нулевая по счету строка содержит коэффициенты  $A_n$ , а первая – коэффициенты  $B_n$ . Для выделения этих коэффициентов выдаваемая матрица транспонируется и из нее выбирается нулевой столбец для коэффициентов  $A_n$ , и первый – для  $B_n$ .

*Листинг 1.23. Программа, вычисляющая коэффициенты Фурье функции.*

```

FC(f,N,L) := 
$$R^{(0)} \leftarrow \begin{pmatrix} \frac{1}{2 \cdot L} \int_{-L}^L f(x) dx \\ 0 \end{pmatrix}$$


$$R^{(n)} \leftarrow \begin{pmatrix} \frac{1}{L} \int_{-L}^L f(x) \cdot \cos\left(\frac{n \cdot \pi \cdot x}{L}\right) dx \\ \frac{1}{L} \int_{-L}^L f(x) \cdot \sin\left(\frac{n \cdot \pi \cdot x}{L}\right) dx \end{pmatrix}$$


$$(R)^T$$

for n ∈ 1..N

$$f(x) := \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ (-x) & \text{if } (-1 \leq x < 0) \\ f(x-2) & \text{if } x > 1 \end{cases}$$

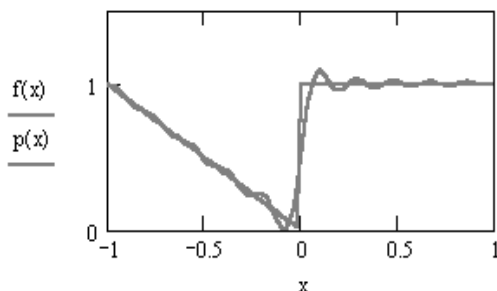
L := 1      N := 10

res := FC(f,N,L)    A := res(0)    B := res(1)    A0 = 0.75


$$p(x) := A_0 + \sum_{n=1}^N \left( A_n \cdot \cos\left(\frac{\pi \cdot n \cdot x}{L}\right) + B_n \cdot \sin\left(\frac{\pi \cdot n \cdot x}{L}\right) \right)$$

h :=  $\frac{L}{50}$     x := -L, -L + h..L

```



## 1.1.4 Функции для проведения регрессии

### 1.1.4.1 Функции для линейной регрессии

Другой широко распространенной задачей обработки данных является представление их совокупности некоторой функцией  $y(x)$ . Задача регрессии заключается в получении параметров этой функции такими, чтобы функция приближала облако исходных точек (заданных векторами  $VX$  и  $VY$ ) с наименьшей среднеквадратичной погрешностью.

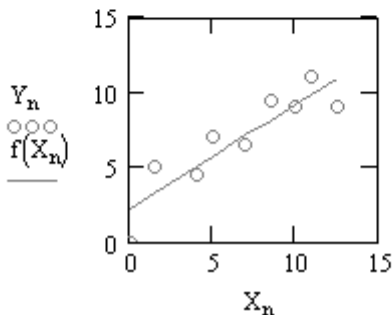
Чаще всего используется линейная регрессия, при которой функция  $y(x)$  имеет вид

$$y(x) = a + b \cdot x$$

и описывает отрезок прямой, как показано на листинге 1.31. К линейной регрессии можно свести многие виды нелинейной регрессии при двупараметрических зависимостях  $y(x)$ .

Листинг 1.31. Линейная регрессия.

```
m :=  $\begin{pmatrix} 0 & 0 \\ 4 & 4.5 \\ 5 & 7 \\ 7 & 6.5 \\ 10 & 9 \\ 11 & 11 \\ 1.5 & 5 \\ 8.5 & 9.5 \\ 12.5 & 9 \end{pmatrix}$       m := csort(m,0)
X := m<0>      Y := m<1>
n := 0..rows(m) - 1
a := intercept(X,Y)
b := slope(X,Y)
f(x) := a + b·x
```



Для проведения линейной регрессии в систему встроено ряд приведенных ниже функций:

- **corr(VX, VY)** — возвращает скаляр — коэффициент корреляции Пирсона;
- **intercpt(VX, VY)** — возвращает значение параметра  $a$  (смещение линии регрессии по вертикали);
- **slope(VX, VY)** — возвращает значение параметра  $b$  (наклона линии регрессии).

#### 1.1.4.2 Функция для линейной регрессии общего вида

В MathCAD реализована возможность выполнения линейной регрессии общего вида. При ней заданная совокупность точек приближается функцией вида:

$$F(x, K_1, K_2, \dots, K_n) = K_1 \cdot F_1(x) + K_2 \cdot F_2(x) + \dots + K_n \cdot F_n(x).$$

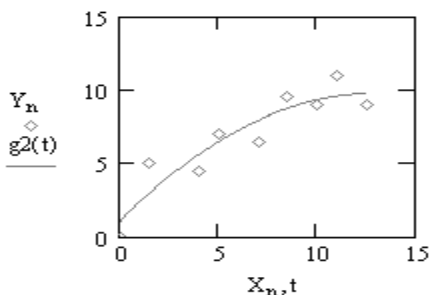
Таким образом, функция регрессии является линейной комбинацией функций  $F_1(x)$ ,  $F_2(x)$ , ...,  $F_n(x)$ , причем сами эти функции могут быть нелинейными, что резко расширяет возможности такой аппроксимации и распространяет ее на нелинейные функции.

Для реализации линейной регрессии общего вида используется функция **linfit(VX, VY, F)**. Эта функция возвращает вектор коэффициентов линейной регрессии общего вида **K**, при котором среднеквадратичная погрешность приближения облака исходных точек, если их координаты хранятся в векторах **VX** и **VY**, оказывается минимальной. Вектор **F** должен содержать функции  $F_1(x)$ ,  $F_2(x)$ , ...,  $F_n(x)$ , записанные в символьном виде, как показано в листинге 1.32.

Листинг 1.32. Линейная регрессия общего вида.

$$F(x) := \begin{pmatrix} 1 \\ x \\ 2 \cdot x^2 - 1 \end{pmatrix} \quad S2 := \text{linfit}(X, Y, F) \quad S2 = \begin{pmatrix} 1.028 \\ 1.347 \\ -0.026 \end{pmatrix}$$

$$g2(t) := F(t) \cdot S2 \quad t := 0, 0.1 \dots 12.5$$





Расположение координат точек исходного массива может быть любым, но вектор **VX** должен содержать координаты, упорядоченные в порядке их возрастания, а вектор **VY** — ординаты, соответствующие абсциссам в векторе **VX**.

1.1.4.3 Функции для одномерной и многомерной полиномиальной регрессии.

Введена в новую версию MathCAD и функция для обеспечения полиномиальной регрессии при произвольной степени полинома регрессии:

**regress(VX,VY, n).**

Она возвращает вектор **VS**, запрашиваемый функцией **interp(VS,VX,VY,x)**, содержащий коэффициенты многочлена  $n$ -й степени, который наилучшим образом приближает “облако” точек с координатами, хранящимися в векторах **VX** и **VY**.

Для вычисления коэффициентов полинома регрессии используется функция **submatrix**.

На практике не рекомендуется делать степень аппроксимирующего полинома выше четвертой - шестой, поскольку погрешности реализации регрессии сильно возрастают.

Функция **regress** создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек, т. е. глобально. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени, — **loess(VX, VY, span)**. Эта функция возвращает используемый функцией **interp(VS,VX,VY,r)** вектор **VS**, дающий наилучшее приближение данных (с координатами точек в векторах **VX** и **VY**) отрезками полиномов второй степени. Аргумент **span**>0 указывает размер локальной области приближаемых данных (рекомендуемое начальное значение - 0,75). На листинге 1.33 показано использование функций **interp** и **loess**.

Чем больше **span**, тем сильнее сказывается сглаживание данных. При больших **span** эта функция приближается к **regress(VX,VY,2)**.

MathCAD позволяет выполнять также МНОГОМЕРНУЮ регрессию, самый типичный случай которой — приближение трехмерных поверхностей. Их можно характеризовать массивом значений высот  $z$ , соответствующих двумерному массиву **Mxy** координат точек  $(x, y)$  на горизонтальной плоскости.

Новых функций для этого не задано. Используются уже описанные выше функции в несколько иной форме (как показано на листинге 1.34):

Функции, используемые для многомерной регрессии:

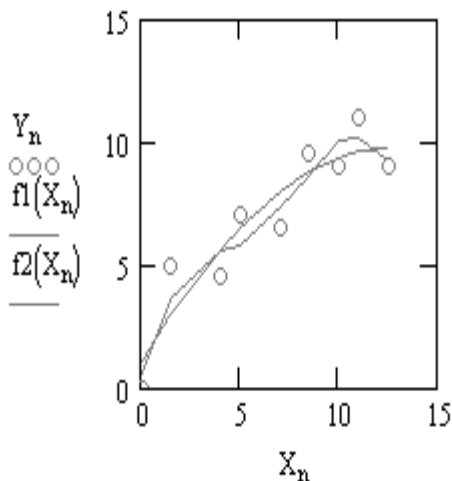
- **regress(Mxy, Vz, n)** — возвращает вектор, запрашиваемый функцией **interp(VS,Mxy,Vz,V)** для вычисления многочлена  $n$ -й степени, который наилучшим образом приближает точки множества **Mxy** и **Vz**. **Mxy** — матрица  $m \times 2$ , содержащая координаты  $x$  и  $y$ .

- **Vz** —  $m$ -мерный вектор, содержащий  $z$ -координаты, соответ-

ствующие  $m$  точкам, указанным в **Mxy**;

Листинг 1.33. Использование функций **interp** и **loess**.

```
vs1 := regress(X,Y,2)  f1(x) := interp(vs1,X,Y,x)
vs2 := loess(X,Y,0.75) f2(x) := interp(vs2,X,Y,x)
```



- **loes(Mxy,Vz,span)** — аналогична **loes(VX,VY, span)**, но в многомерном случае;
- **interp (VS,Mxy,Vz,V)** — возвращает значение  $z$  по заданным векторам **VS** (создается функциями **regress** или **loess**) и **Mxy**, **Vz** и **V** (вектор координат  $x$  и  $y$  заданной точки, для которой находится  $z$ ):

$$VS1 := \text{loess}(Mxy, Vz, 0.85)$$

$$Z1_{i,j} := \text{interp} \left[ VS1, Mxy, Vz, \begin{pmatrix} x_i \\ y_j \end{pmatrix} \right]$$

Листинг 1.34. Использование функций для многомерной регрессии.

```

Mxy :=  $\begin{pmatrix} 2 & 2 \\ 2 & 3 \\ 3 & 2 \\ 3 & 3 \\ 1 & 1 \\ 1 & 4 \\ 4 & 1 \\ 4 & 4 \\ 0 & 0 \\ 0 & 5 \\ 5 & 0 \\ 5 & 5 \end{pmatrix}$ 
VZ :=  $\begin{pmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 4 \\ 3 \\ 5 \\ 2 \\ 3 \\ 2 \\ 1 \end{pmatrix}$ 
VS := regress(Mxy,VZ,3)

i := 0..50    j := 0..50

xi :=  $\frac{i}{10}$     yj :=  $\frac{j}{10}$ 

Z3i,j := interp  $\left[ VS, Mxy, VZ, \begin{pmatrix} x_i \\ y_j \end{pmatrix} \right]$ 
X := Mxy<0>    Y := Mxy<1>    Z := VZ

```

На рис. 1.12 и 1.13 приведены результаты использования функций для многомерной регрессии.

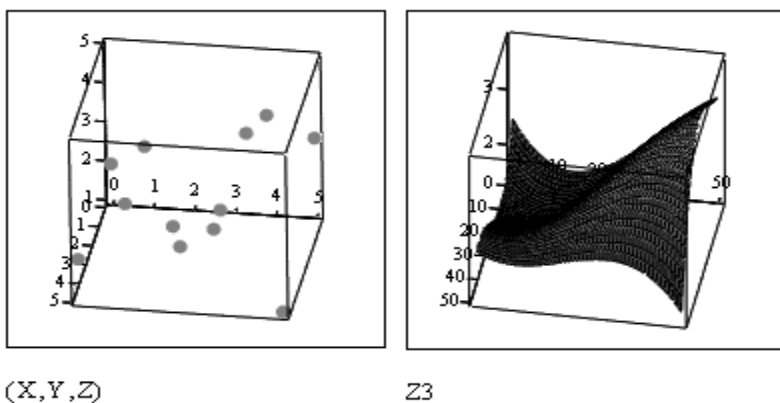
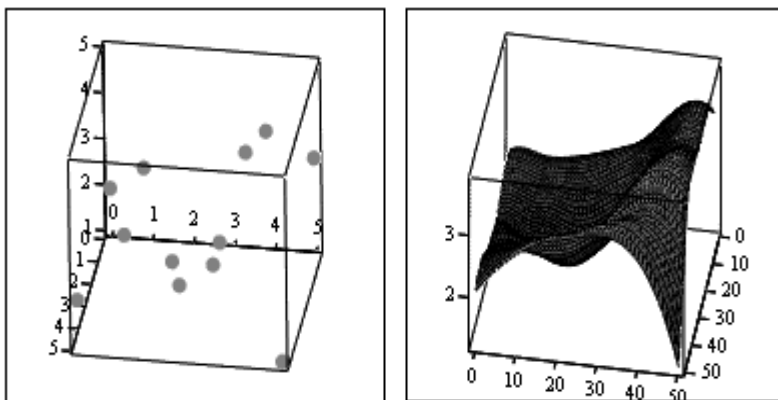


Рисунок 1.12 – Примеры результатов использования функции многомерной регрессии



(X,Y,Z)

Z1

Рисунок 1.13 – Примеры результатов использования функции многомерной регрессии

#### 1.1.4.4 Функция для нелинейной регрессии общего вида.

Под нелинейной регрессией общего вида подразумевается нахождение вектора **K** параметров произвольной функции  $F(x, K1, K2, ..., Kn)$ , при котором обеспечивается минимальная среднеквадратичная погрешность приближения облака исходных точек.

Для проведения нелинейной регрессии общего вида используется функция **genfit(VX, VY, VS, F)**. Эта функция возвращает вектор **K** параметров функции **F**, дающий минимальную среднеквадратичную погрешность приближения функцией  $F(x, K1, K2, ..., Kn)$  исходных данных, как показано на рис. 1.14.

**F** должен быть вектором с символьными элементами, содержащими уравнение исходной функции и ее производных по всем параметрам. Вектор **VS** должен содержать начальные значения элементов вектора **K**, необходимые для решения системы нелинейных уравнений регрессии итерационным методом.

## 1.2 Порядок выполнения работы

### 1.2.1 Задания для выполнения работы

1. Составьте программу в MathCAD по соответствующему варианту задания, приведенному в пункте 1.2.2.1.
2. Выполните регрессию экспериментальных данных по варианту задания, приведенного в пункте 1.2.2.2.

$$\begin{aligned}
 & \text{XX} := \begin{pmatrix} 1.5 \\ 4 \\ 5 \\ 7 \\ 8.5 \\ 10 \\ 11 \\ 12.5 \end{pmatrix} \quad \text{YY} := \begin{pmatrix} 5 \\ 4.5 \\ 7 \\ 6.5 \\ 9 \\ 11 \\ 5 \\ 9.5 \end{pmatrix} \quad \begin{aligned} & \text{ff}(x, r, w) := r \cdot \ln(w \cdot x) \\ & \frac{d}{dr} \text{ff}(x, r, w) \rightarrow \ln(w \cdot x) \\ & \frac{d}{dw} \text{ff}(x, r, w) \rightarrow \frac{r}{w} \end{aligned} \\
 & \text{F}(x, U) := \begin{pmatrix} U_0 \cdot \ln(U_1 \cdot x) \\ \ln(U_1 \cdot x) \\ U_0 \\ U_1 \end{pmatrix} \quad \text{VG} := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
 & \quad \quad \quad k := 0 \dots \text{rows}(\text{XX}) - 1 \\
 & U := \text{genfit}(\text{XX}, \text{YY}, \text{VG}, \text{F}) \quad U = \begin{pmatrix} 2.051 \\ 5.288 \end{pmatrix} \\
 & y(t) := U_0 \cdot \ln(U_1 \cdot t)
 \end{aligned}$$

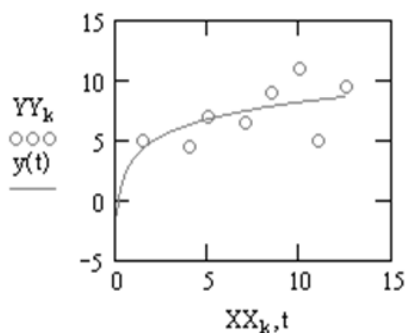


Рисунок 1.13 - Результат использования функций для нелинейной регрессии общего вида.

## 1.2.2 Варианты заданий

### 1.2.2.1 Варианты заданий для составления программ

1. а) Составить в MathCAD программу, выводящую в массив квадраты всех натуральных чисел от 0 до заданного натурального **n**.

б)  $\gamma$  - функция Вейерштрасса определена соотношением:

$$\gamma(z) = \frac{1}{z^2} + \sum_{m_1, m_2} \left( \frac{1}{(z-w)^2} - \frac{1}{w^2} \right)$$

где  $w = m_1 w_1 + m_2 w_2$ . Суммирование производится по всем значениям  $m_1, m_2 = 0, \pm 1, \pm 2, \dots$ , за исключением  $m_1 = m_2 = 0$ .

Составить в MathCAD программу вычисления  $\gamma$  - функции Вейерштрасса для  $w_1 = 2.3$  и  $w_2 = 3.1$  с точностью  $\varepsilon = 5 \cdot 10^{-4}$ . Построить график функции  $\gamma(z)$  в интервале  $z \in [0, 15]$ .

с) Составить в MathCAD программу преобразования  $\mu, v, \lambda$  – обших эллипсоидальных координат с параметрами  $k$  и  $c$  в декартовы координаты  $x, y$ , и  $z$  с помощью преобразования:

$$x = \pm \frac{c}{k} \mu v \lambda, \quad y = \pm \frac{c}{k} \frac{1}{\sqrt{1-k^2}} \sqrt{(k^2 - \mu^2)(v^2 - k^2)(\lambda^2 - k^2)},$$

$$z = \pm c \frac{1}{\sqrt{1-k^2}} \sqrt{(1 - \mu^2)(1 - v^2)(\lambda^2 - 1)},$$

Причем должны выполняться соотношения:  $\lambda > 1 > v > k > \mu > 0$ . Построить в координатах  $(x, y, z)$  поверхности постоянных значений эллипсоидальных координат:

$$\lambda(\mu, v) = \text{const}, \mu(\lambda, v) = \text{const}, v(\mu, \lambda) = \text{const}.$$

2. а) Составить в MathCAD программу, создающую массив, содержащий числа, являющиеся результатом разложения на простые множители заданного натурального числа  $n > 0$ .

б) Неполный эллиптический интеграл 1-го рода имеет вид:

$$F(k, \varphi) = \int_0^\varphi \frac{d\psi}{\sqrt{1 - k^2 \sin^2 \psi}},$$

где  $0 < k^2 < 1$ . Составить в MathCAD программу вычисления неполного эллиптического интеграла 1-го рода. Построить поверхность значений  $F(k, \varphi)$  в диапазоне  $k \in (-1, 1)$  и  $\varphi \in [0, \pi]$ .

с) Конические координаты  $u, v$  связаны с декартовыми координатами  $x, y$ , и  $z$  с помощью преобразования:

$$x + iy = \frac{2r}{1 + |f|^2} f(u + iv), \quad z = r \frac{1 - |f|^2}{1 + |f|^2},$$

где  $f(u + iv)$  – некоторая функция комплексной переменной. Построить в координатах  $(x, y, z)$  поверхности постоянных значений конических координат:

$$r(u, v) = \text{const}, u(r, v) = \text{const}, v(r, u) = \text{const},$$

$$\text{если функция преобразования имеет вид } f(u + iv) = e^{\sqrt{u} + i g(v)}.$$

3. а) Последовательность Фибоначчи определяется так:  $a(0) = 1$ ,  $a(1) = 1$ ,  $a(k) = a(k-1) + a(k-2)$  при  $k \geq 2$ . Составьте в MathCAD программу для вычисления чисел Фибоначчи для любого  $k \geq 0$ . Результат представить в виде массива значений чисел Фибоначчи в диапазоне от  $a(0)$  до  $a(k)$ .

б)  $\Gamma(x)$  – функция (факториал) может быть представлена для больших  $x$  в виде бесконечного произведения:

$$\frac{1}{\Gamma(x)} = x \cdot e^{Cx} \prod_{k=1}^{\infty} \left(1 + \frac{x}{k}\right) e^{-\frac{x}{k}}, \quad C = 0.577216 - \text{постоянная Эйлера.}$$

Составить в MathCAD программу вычисления  $\Gamma$  – функции ( $1/\Gamma(x)$ ) с точностью  $\varepsilon = 2 \cdot 10^{-6}$ . Построить график функции  $1/\Gamma(x)$  в диапазоне  $x \in [-4.1, 4]$ .

с) Пространственные биполярные координаты  $u, v, \varphi$  связаны с декартовыми координатами  $x, y, z$  с помощью преобразования, приведенного ниже. Составить в MathCAD программу, преобразования координат точки  $(u, v, \varphi)$  в  $(x, y, z)$ . Построить в координатах  $(x, y, z)$  поверхность постоянных значений пространственных биполярных координат:

$$u(v, \varphi) = \text{const}, v(u, \varphi) = \text{const}, \varphi(u, v) = \text{const}.$$

$$x = \frac{\alpha \sin(v)}{ch(u) + \cos(v)} \cos(\varphi) \quad -\infty < u < \infty$$

$$y = \frac{\alpha \sin(v)}{ch(u) + \cos(v)} \sin(\varphi) \quad 0 \leq v \leq \pi,$$

$$z = \frac{\alpha sh(v)}{ch(u) + \cos(v)} \quad 0 \leq \varphi \leq 2\pi$$

где  $\pm \alpha$  – положение полюсов на оси  $z$ .

4. а) Составьте в MathCAD программу для вычисления факториала.

б) Функция Эрмита  $H_n(x)$  с индексом  $n$  выражается с помощью ряда:

$$H_n(x) = \sum_{k=0}^n c_k x^k, \quad \text{где } c_0 = A, \quad c_1 = Bx; \quad c_{k+2} = c_k \frac{2(n-k)}{(n+1)(n+2)}.$$

$A$  и  $B$  – произвольные постоянные (индекс  $n$  не обязательно целое число). При целом  $n$  функция Эрмита превращается в полином степени  $n$ . Составьте в MathCAD программу для вычисления функции Эрмита с произвольным индексом с точностью  $\varepsilon = 1 \cdot 10^{-6}$ . Параметры  $A = -1$ ;  $B = 2$ . Построить график функции  $H_n(x)$  в диапазоне  $x \in [-1, 1]$  для индексов  $n = 0.7, 0.9, 1.5, 2, 4.2$ .

с) Тороидальные координаты  $u, v, \varphi$  (семейство  $u$  – торы и семейство сферических  $v$  – сегментов, соответственно заключающих внутри

себя и проходящих через окружность радиуса  $\alpha$  с центром в начале координат, лежащую в  $\mathbf{x}, \mathbf{y}$  – плоскости) связаны с декартовыми координатами  $\mathbf{x}, \mathbf{y}$ , и  $\mathbf{z}$  с помощью преобразования, приведенного ниже. Составить в MathCAD программу, преобразования координат точки  $(\mathbf{u}, \mathbf{v}, \varphi)$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений то-roidальных координат:

$$\mathbf{u}(\mathbf{v}, \varphi) = \text{const}, \mathbf{v}(\mathbf{u}, \varphi) = \text{const}, \varphi(\mathbf{u}, \mathbf{v}) = \text{const}.$$

$$x = \frac{\alpha \cdot sh(u)}{ch(u) + \cos(v)} \cos(\varphi) \quad 0 \leq u < \infty$$

$$y = \frac{\alpha sh(u)}{ch(u) + \cos(v)} \sin(\varphi) \quad 0 \leq v < 2\pi.$$

$$z = \frac{\alpha \sin(v)}{ch(u) + \cos(v)} \quad 0 \leq \varphi < 2\pi$$

5. а) Составьте в MathCAD программу для вычисления функции  $S(n) = 1/0! + 1/1! + \dots + 1/n!$ , где  $n$  – натуральное число.

б)  $\Psi$ -функция может быть определена с помощью соотношения:

$$\Psi(x) = \sum_{k=1}^{\infty} \left( \frac{1}{k} - \frac{1}{x+k} \right) - C,$$

где  $C = 0.577216$  – постоянная Эйлера. Составьте в MathCAD программу для вычисления  $\Psi$ -функции с точностью  $\varepsilon = 1 \cdot 10^{-6}$ . Построить график  $\Psi$ -функции  $\Psi(\mathbf{x})$  в диапазоне  $\mathbf{x} \in [-0.5, 10]$ .

с) Координаты сплюсненного эллипсоида вращения (со фокусные однополостные  $u$  – гиперboloиды вращения и сплюсненные  $v$  – эллипсоиды вращения).

$$u + iv = \arcsin \frac{\rho + iz}{\alpha},$$

$$\sin u = \frac{s_1 - s_2}{2\alpha}, \quad chv = \frac{s_1 + s_2}{2\alpha}, \quad \varphi = \arctg \frac{y}{x},$$

где  $s_1 = \sqrt{z^2 + (\rho + \alpha)^2}$ ,  $s_2 = \sqrt{z^2 + (\rho - \alpha)^2}$  представляют собой соответственно наибольшее и наименьшее расстояние от точки  $\mathbf{x}, \mathbf{y}, \mathbf{z}$  до окружности радиуса  $\alpha$  с центром в начале координат, лежащей в  $\mathbf{x}, \mathbf{y}$  – плоскости. При отрицательном  $\mathbf{z}$  следует для  $s_1$  и  $s_2$  брать отрицательные значения.



$$\begin{aligned}
x &= \alpha \sin u \cdot \operatorname{ch} v \cdot \cos \varphi, & 0 \leq u \leq \pi, \\
y &= \alpha \sin u \cdot \operatorname{ch} v \cdot \sin \varphi, & 0 \leq v < \infty, \\
z &= \alpha \cos u \cdot \operatorname{sh} v, & 0 \leq \varphi < 2\pi, \\
\rho &= \sqrt{x^2 + y^2} = \alpha \sin u \cdot \operatorname{ch} v.
\end{aligned}$$

Составить в MathCAD программу, преобразования координат точки  $(\mathbf{u}, \mathbf{v}, \varphi)$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений тороидальных координат:

$$\mathbf{u}(\mathbf{v}, \varphi) = \text{const}, \mathbf{v}(\mathbf{u}, \varphi) = \text{const}, \varphi(\mathbf{u}, \mathbf{v}) = \text{const}.$$

6. а) Составьте в MathCAD программу для вычисления  $C$  – постоянной Эйлера с точностью  $\varepsilon = 1.5 \cdot 10^{-3}$ , если  $C = \lim_{n \rightarrow \infty} \left( \sum_{k=1}^n \left( \frac{1}{k} \right) - \ln(n) \right)$

б) Функция Лагерра  $L_n(x)$  может быть определена с помощью соотношения:

$$L_0(x) = 1; \quad L_1(x) = 1 - x; \quad L_{n+1}(x) = (2n + 1 - x)L_n(x) - n^2 L_{n-1}(x).$$

Составьте в MathCAD программу для вычисления функции Лагерра  $L_n(x)$ . Построить график функции Лагерра  $L_n(x) / n!$  в диапазоне  $x \in [-2.5, 20]$  при  $n = 3, 4, 5$ .

с) Параболические координаты вращения:

$$\begin{aligned}
\frac{u + iv}{\sqrt{2}} &= \sqrt{z + i\rho}, \quad \rho = \sqrt{x^2 + y^2} \\
u^2 &= r + z & x &= uv \cos(\varphi) & 0 \leq u < \infty \\
v^2 &= r - z & y &= uv \sin(\varphi) & 0 \leq v < \infty \\
\varphi &= \arctg\left(\frac{y}{x}\right) & z &= \frac{u^2 - v^2}{2} & 0 \leq \varphi < 2\pi \\
& & & & r &= \sqrt{z^2 + \rho^2}
\end{aligned}$$

Составить в MathCAD программу, преобразования координат точки  $(\mathbf{u}, \mathbf{v}, \varphi)$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений параболических координат вращения:

$$\mathbf{u}(\mathbf{v}, \varphi) = \text{const}, \mathbf{v}(\mathbf{u}, \varphi) = \text{const}, \varphi(\mathbf{u}, \mathbf{v}) = \text{const}.$$

7. а) Составьте в MathCAD программу для вычисления  $\zeta(x)$  – функции Римана с точностью  $\varepsilon = 10^{-6}$ :

$$\zeta(x) = \sum_{k=1}^{\infty} \frac{1}{k^x}.$$

Построить график  $\zeta(x)$  – функции Римана в диапазоне  $x \in [1, 3]$ .

б) Составьте в MathCAD программу для вычисления гипергеометрической функции  $F(\alpha, \gamma, x)$  с точностью  $\varepsilon = 10^{-6}$ :

$$F(\alpha, \gamma, x) = 1 + \frac{\alpha}{\gamma} x + \frac{\alpha(\alpha+1)}{\gamma(\gamma+1)} \frac{x^2}{2!} + \frac{\alpha(\alpha+1)(\alpha+2)}{\gamma(\gamma+1)(\gamma+2)} \frac{x^3}{3!} + \dots$$

Построить график функции  $F(\alpha, \gamma, x)$  в диапазоне  $x \in [-1; 3]$  при различных комбинациях значений параметров  $\alpha$  и  $\gamma$  из диапазона  $[0.1; 3]$ .

с) Круговые цилиндрические координаты  $u, v, z$ :

$$u = \ln\left(\sqrt{x^2 + y^2}\right), \quad x = e^u \cos(v) \quad -\infty < u < +\infty$$

$$v = \arctg\left(\frac{y}{x}\right) \quad y = e^u \sin(v) \quad 0 \leq v < 2\pi$$

$$z \quad z \quad -\infty < z < +\infty$$

Составить в MathCAD программу, преобразования координат точки  $(u, v, z)$  в  $(x, y, z)$ . Построить в координатах  $(x, y, z)$  поверхности постоянных значений параболлических координат вращения:

$$u(v, z) = \text{const}, v(u, z) = \text{const}, z(u, v) = \text{const}.$$

8. а) Составьте в MathCAD программу для вычисления суммы ряда разложения интеграла ошибок Гаусса с точностью  $\varepsilon = 10^{-6}$ :

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \left( \frac{x}{1} - \frac{x^3}{1 \cdot 3} + \frac{x^5}{2 \cdot 5} - \frac{x^7}{3 \cdot 7} + \dots - (-1)^{2n+1} \frac{x^{2n+1}}{n!(2n+1)} + \dots \right)$$

Построить график  $\Phi(x)$  – интеграла ошибок Гаусса в диапазоне  $x \in [-1, 1]$ .

б) Составьте в MathCAD программу для вычисления функции Чебышева  $T_n(x)$  с точностью  $\varepsilon = 10^{-6}$ :

$$T_n(x) = \sum_{k=0}^{\infty} c_k x^k, \quad \text{где } c_{k+2} = \frac{(k+n)(k-n)}{(k+1)(k-1)} c_k,$$

$$c_0 = A, \quad c_1 = B.$$

Построить график функции  $T_n(x)$  в диапазоне  $x \in [-1; 3]$  при различных значениях параметров  $n, A$  и  $B$  из диапазона  $[-2; 5]$ .

с) Координаты вытянутого эллипсоида вращения  $u, v, \varphi$  (софокусные двуполостные  $u$ - гиперboloиды вращения и вытянутые  $v$ - эллипсоиды вращения):

$$\sin(u) = \frac{s_1 - s_2}{2\alpha}, \quad \text{ch}(v) = \frac{s_1 + s_2}{2\alpha}, \quad \varphi = \arctg\left(\frac{y}{x}\right),$$

$$\text{где } s_1 = \sqrt{x^2 + y^2 + (z + \alpha)^2}, \quad s_2 = \sqrt{x^2 + y^2 + (z - \alpha)^2}$$

– расстояния точки  $x, y, z$  от фокусов, лежащих на оси  $z$  в точках  $z_1 = -\alpha$  и  $z_2 = +\alpha$

$$x = \alpha \cos u \cdot sh(v) \cdot \cos \varphi, \quad -\frac{\pi}{2} \leq u \leq +\frac{\pi}{2}$$

$$y = \alpha \cos u \cdot sh(v) \cdot \sin \varphi \quad 0 \leq v < \infty$$

$$z = \alpha \sin u \cdot ch(v) \quad 0 \leq \varphi < 2\pi$$

$$\rho = \sqrt{x^2 + y^2} = \alpha \cos u \cdot sh(v)$$

Составить в MathCAD программу, преобразования координат точки  $(u, v, \varphi)$  в  $(x, y, z)$ . Построить в координатах  $(x, y, z)$  поверхности постоянных значений параболических координат вращения:

$$u(v, \varphi) = \text{const}, v(u, \varphi) = \text{const}, \varphi(u, v) = \text{const}.$$

9. а) Составьте в **MathCAD** программу для вычисления суммы ряда разложения интегральной показательной функции  $Ei(x)$  с точностью  $\varepsilon = 10^{-6}$ :

$$Ei(x) = \int_{-\infty}^{-x} \frac{e^{-t}}{t} dt = C + \ln x + \frac{x}{1 \cdot 1!} + \frac{x^2}{2 \cdot 2!} + \dots$$

для всех  $x > 0$ , где  $C = 0,577216$  – постоянная Эйлера. Построить график интегральной показательной функции  $Ei(x)$  в диапазоне  $x \in [1, 3]$ .

б) Составьте в MathCAD программу для вычисления цилиндрической функции Бесселя  $J_n(x)$  с точностью  $\varepsilon = 10^{-6}$ :

$$J_n(x) = Ax^n \left[ 1 - \frac{x^2}{2(2n+2)} + \frac{x^4}{2 \cdot 4(2n+2)(2n+4)} - \frac{x^6}{2 \cdot 4 \cdot 6(2n+2)(2n+4)(2n+6)} + \dots \right] + Bx^{-n} \left[ 1 + \frac{x^2}{2(2n-2)} + \frac{x^4}{2 \cdot 4(2n-2)(2n-4)} + \dots \right].$$

Построить график функции  $J_n(x)$  в диапазоне  $x \in [-1; 3]$  при различных значениях параметров  $n$ ,  $A$  и  $B$  из диапазона  $[-2; 5]$ .

с) Параболические цилиндрические координаты  $u, v, z$  (софокусные  $u$ - и  $v$ - параболы с общей осью симметрии к ним добавлена не зависящая от них координата  $z$ ):

$$\frac{u+iv}{\sqrt{2}} = \sqrt{x+iy},$$

$$u = \sqrt{r+x}, \quad \text{где } r = \sqrt{x^2 + y^2} \quad (\text{расстояние от фокуса } x=0, y=0),$$

$$v = \sqrt{r-x},$$

$$\left. \begin{aligned} x &= \frac{u^2 - v^2}{2}, & 0 \leq u < \infty \\ y &= \pm u \cdot v, & 0 \leq v < \infty \end{aligned} \right\} \quad \begin{aligned} & \text{(для данных значений } u, v \\ & \text{получаются два значения } y!). \end{aligned}$$

Составить в MathCAD программу, преобразования координат точки  $(\mathbf{u}, \mathbf{v}, \mathbf{z})$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений параболических цилиндрических координат:

$$\mathbf{u}(\mathbf{v}, \mathbf{z}) = \text{const}, \mathbf{v}(\mathbf{u}, \mathbf{z}) = \text{const}, \mathbf{z}(\mathbf{u}, \mathbf{v}) = \text{const}.$$

10. а) Составьте в MathCAD программу для вычисления суммы ряда разложения интегрального синуса  $\text{Si}(x)$  с точностью  $\varepsilon = 10^{-6}$ :

$$\text{Si}(x) = \int_{-\infty}^x \frac{\sin t}{t} dt = x - \frac{x^3}{3 \cdot 3!} + \frac{x^5}{5 \cdot 5!} - \dots$$

для всех  $x$ . Построить график интегральной показательной функции  $\text{Si}(x)$  в диапазоне  $x \in [-4, 4]$ .

б) Запишите в MathCAD выражение для вычисления неполного эллиптического интеграла 2-го рода:

$$E(k, \varphi) = \int_0^{\varphi} \sqrt{1 - k^2 \sin^2 \psi} d\psi.$$

Построить график функции  $E(k, \varphi)$  в диапазоне  $k \in [-1; 1]$  при различных значениях параметров  $\varphi$  из диапазона  $[0.2; 5]$ .

с) Эллиптические цилиндрические координаты  $\mathbf{u}, \mathbf{v}, \mathbf{z}$  (софокусные  $\mathbf{u}$ - гиперболы и  $\mathbf{v}$ - эллипсы к ним добавлена не зависящая от них координата  $\mathbf{z}$ ):

$$u + iv = \arcsin \frac{x + iy}{\alpha},$$

$$\sin u = \frac{s_1 - s_2}{2\alpha}, \quad chv = \frac{s_1 + s_2}{2\alpha},$$

где  $s_1 = \sqrt{(x + \alpha)^2 + y^2}$ ,  $s_2 = \sqrt{(x - \alpha)^2 + y^2}$  -расстояние точки  $x, y$  от фокусов лежащих на  $x$ -оси в точках  $x_1 = -\alpha$ , и  $x_2 = \alpha$ ,

$$\left. \begin{aligned} x &= \alpha \sin u \cdot chv, & 0 \leq u < 2\pi \\ y &= \pm \cos u \cdot shv, & 0 \leq v < \infty \end{aligned} \right\} \begin{aligned} &\text{(для данных значений } u, v \\ &\text{получаются два значения } y!). \end{aligned}$$

Составить в MathCAD программу, преобразования координат точки  $(\mathbf{u}, \mathbf{v}, \mathbf{z})$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений эллиптических цилиндрических координат:

$$\mathbf{u}(\mathbf{v}, \mathbf{z}) = \text{const}, \mathbf{v}(\mathbf{u}, \mathbf{z}) = \text{const}, \mathbf{z}(\mathbf{u}, \mathbf{v}) = \text{const}.$$

11. а) Составьте в MathCAD программу для вычисления суммы ряда для  $\ln(1+x)$  с точностью  $\varepsilon = 10^{-6}$ :

$$\ln(1+x) = \frac{x}{1} - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$

Построить график зависимости  $\ln(1+x)$  в диапазоне  $x \in [-0.1, 1]$ .

б) Запишите в MathCAD выражение для вычисления общей сферической функции  $\mathbf{Y}_n(\theta, \varphi)$ :

$$Y_n(\vartheta, \varphi) = \sum_m C_{nm} Y_n^m(\vartheta, \varphi), \quad \text{где } Y_n^m(\vartheta, \varphi) = P_n^m(\cos(\vartheta)) e^{im\varphi},$$

$$P_{n+1}^m(x) = \frac{(2n+1)xP_n^m(x) - (n+m)P_{n-1}^m(x)}{n-m+1}, \quad P_n^0(x) = P_n(x),$$

$$P_n^m(x) = P_n^{-m}(x) = \left(\sqrt{1-x^2}\right)^m \frac{d^m}{dx^m} P_n(x),$$

$$P_{n+1}(x) = \frac{(2n+1) \cdot xP_n(x) - nP_{n-1}(x)}{n+1}, \quad P_0(x) = 1, \quad P_1(x) = x.$$

Построить трехмерный график действительной и мнимой частей общей сферической функции  $\mathbf{Y}_n(\boldsymbol{\theta}, \boldsymbol{\varphi})$  в диапазоне координат  $\boldsymbol{\theta} \in [-\pi/2; \pi/2]$  и  $\boldsymbol{\varphi} \in [0; 2\pi]$  для сферы радиуса  $r = 1$  при различных значениях параметра  $\mathbf{n}$  из диапазона  $[2; 5]$ .

с) Составить в MathCAD программу преобразования  $\boldsymbol{\mu}, \mathbf{v}, \boldsymbol{\lambda}$  – общих эллипсоидальных координат с параметрами  $\mathbf{k}$  и  $\mathbf{c}$  в декартовы координаты  $\mathbf{x}, \mathbf{y}$ , и  $\mathbf{z}$  с помощью преобразования:

$$x = \pm \frac{c}{k} \mu \nu \lambda, \quad y = \pm \frac{c}{k} \frac{1}{\sqrt{1-k^2}} \sqrt{(k^2 - \mu^2)(\nu^2 - k^2)(\lambda^2 - k^2)},$$

$$z = \pm c \frac{1}{\sqrt{1-k^2}} \sqrt{(1-\mu^2)(1-\nu^2)(\lambda^2 - 1)},$$

Причем должны выполняться соотношения:  $\boldsymbol{\lambda} > \mathbf{1} > \mathbf{v} > \mathbf{k} > \boldsymbol{\mu} > \mathbf{0}$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений эллипсоидальных координат при различных значениях параметров  $\mathbf{k}$  и  $\mathbf{c}$ :

$$\boldsymbol{\lambda}(\boldsymbol{\mu}, \mathbf{v}) = \text{const}, \boldsymbol{\mu}(\boldsymbol{\lambda}, \mathbf{v}) = \text{const}, \mathbf{v}(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \text{const}.$$

12. а) Составьте в MathCAD программу для вычисления  $\binom{n}{r}$  числа сочетаний  $r$  элементов из  $n$  различных элементов.

$$C_n^r = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Построить график зависимости  $f(x) = \sum_{k=0}^n C_n^k x^{n-k}$  в диапазоне  $\mathbf{x} \in [0, 2]$  при различных значениях параметра  $\mathbf{n}$  из диапазона  $[2; 5]$ .

б) Запишите в MathCAD выражение для вычисления функции Чебышева  $U_n(\mathbf{x})$  с точностью  $\boldsymbol{\varepsilon} = 10^{-6}$ :

$$U_n(x) = \sqrt{1-x^2} Q_{n-1}(x), \quad \text{кроме } U_0(x) = \arcsin x;$$

$$Q_{n-1}(x) = \frac{1}{n} T_n'(x)$$

$$T'_n(x) = \sum_{k=0}^{\infty} c_k k \cdot x^{k-1}, \quad \text{где } c_{k+2} = \frac{(k+n)(k-n)}{(k+1)(k-1)} c_k,$$

$$c_0 = A, \quad c_1 = B.$$

Построить график функции  $U_n(\mathbf{x})$  в диапазоне в диапазоне  $\mathbf{x} \in [-1; 1]$  при различных значениях параметров  $\mathbf{n}$ ,  $\mathbf{A}$  и  $\mathbf{B}$  из диапазона  $[-2; 5]$ .

с) Вращательно-симметричные координаты  $\mathbf{u}, \mathbf{v}, \varphi$ :

$$u = u(z, \rho), \quad v = v(z, \rho), \quad x = \rho \cos \varphi, \quad y = \rho \sin \varphi,$$

$$\rho = \sqrt{x^2 + y^2}, \quad \varphi = \arctg \frac{y}{x}.$$

$$z + i\rho = f(u + iv),$$

где  $f(u + iv) = \ln(\sqrt{u + iv})$  – аналитическая функция.

Составить в MathCAD программу, преобразования координат точки  $(\mathbf{u}, \mathbf{v}, \varphi)$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений вращательно-симметричных координат:

$$\mathbf{u}(\mathbf{v}, \varphi) = \text{const}, \mathbf{v}(\mathbf{u}, \varphi) = \text{const}, \varphi(\mathbf{u}, \mathbf{v}) = \text{const}.$$

13. а) Составьте в MathCAD программу для вычисления суммы ряда для интегрального косинуса  $Ci(\mathbf{x})$  с точностью  $\varepsilon = 10^{-6}$ :

$$Ci(x) = -\int_x^{\infty} \frac{\cos t}{t} dt = C + \ln x - \frac{x^2}{2 \cdot 2!} + \frac{x^4}{4 \cdot 4!} - \dots \text{ (для всех } x \neq 0),$$

где  $C = 0.577216$  – постоянная Эйлера. Построить график зависимости  $Ci(\mathbf{x})$  в диапазоне  $\mathbf{x} \in [0.7, 2]$ .

б) Запишите в MathCAD выражение для вычисления сферических функций Лежандра  $Q_n(\mathbf{x})$  с точностью  $\varepsilon = 10^{-6}$ :

$$Q_0(x) = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots$$

$$Q_1(x) = -1 + x^2 + \frac{x^4}{3} + \frac{x^6}{5} + \dots$$

$$Q_{n+1}(x) = \frac{2n+1}{n+1} x Q_n(x) - \frac{n}{n+1} Q_{n-1}(x)$$

Построить график функции  $Q_n(\mathbf{x})$  в диапазоне в диапазоне  $\mathbf{x} \in [-1; 1]$  при различных значениях параметров  $\mathbf{n}$  из диапазона  $[2; 5]$ .

с) Конические координаты  $\mathbf{r}, \mathbf{u}, \mathbf{v}$ . Их получают при помощи стереографической проекции, проектируя двумерную ортогональную систему координат  $\mathbf{u}, \mathbf{v}$  на сферу радиуса 1 и присоединяя радиальную координату  $r$ , т.е. при помощи соотношений

$$\frac{x + iy}{r + z} = f(u + iv), \quad r = \sqrt{x^2 + y^2 + z^2},$$

или в явном виде:

$$x+iy = \frac{2r}{1+|f|^2} f(u+iv), \quad z = r \frac{1-|f|^2}{1+|f|^2},$$

где  $f(u+iv) = \sqrt{\ln(u+iv)}$  – аналитическая функция.

Составить в MathCAD программу, преобразования координат точки  $(\mathbf{r}, \mathbf{u}, \mathbf{v})$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений конических координат:

$$\mathbf{u}(\mathbf{v}, \mathbf{r}) = \text{const}, \mathbf{v}(\mathbf{u}, \mathbf{r}) = \text{const}, \mathbf{r}(\mathbf{u}, \mathbf{v}) = \text{const}.$$

14. а) Составьте в MathCAD программу для вычисления суммы ряда для интеграла Френеля  $\mathbf{C}(\mathbf{x})$  с точностью  $\varepsilon = 10^{-6}$ :

$$C(x) = \frac{1}{2\sqrt{x}} \int_0^x \frac{\sin t}{\sqrt{t}} dt = \frac{1}{\sqrt{x}} \int_0^{\sqrt{x}} \cos t^2 dt = 1 - \frac{x^2}{2!5} + \frac{x^4}{4!9} - \frac{x^6}{6!13} + \dots$$

Построить график зависимости  $\mathbf{C}(\mathbf{x})$  в диапазоне  $\mathbf{x} \in [0.7, 2]$ .

б) Запишите в MathCAD выражение для вычисления общей гипергеометрической функции  $\mathbf{F}(\alpha, \beta, \gamma, \mathbf{x})$  с точностью  $\varepsilon = 10^{-6}$ :

$$F(\alpha, \beta, \gamma, x) = 1 + \frac{\alpha \cdot \beta}{\gamma} x + \frac{\alpha(\alpha+1)\beta(\beta+1)}{\gamma(\gamma+1)} \frac{x^2}{2!} + \\ + \frac{\alpha(\alpha+1)(\alpha+2)\beta(\beta+1)(\beta+2)}{\gamma(\gamma+1)(\gamma+2)} \frac{x^3}{3!} + \dots$$

Построить график функции  $\mathbf{F}(\alpha, \beta, \gamma, \mathbf{x})$  в диапазоне в диапазоне  $\mathbf{x} \in [-1; 1]$  при различных нецелых значениях параметров  $\alpha, \beta$  и  $\gamma$  из диапазона [2; 5].

с) Общие неортогональные системы координат  $\mathbf{u}, \mathbf{v}, \mathbf{z}$ :

$$u = \xi(x, y) \quad v = \zeta(x, y) \quad z = z$$

где  $\xi(x, y)$  и  $\zeta(x, y)$  – аналитические функции.

Составить в MathCAD программу, преобразования координат точки  $(\mathbf{u}, \mathbf{v}, \mathbf{z})$  в  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ , если  $\xi(x, y) = \sqrt{x \cdot y}$  и  $\zeta(x, y) = \sqrt{2x - 3y}$ . Найти зависимость (решить аналитически (в символьном виде) соответствующую систему уравнений)  $\mathbf{x} = \mathbf{X}(\mathbf{u}, \mathbf{v})$  и  $\mathbf{y} = \mathbf{Y}(\mathbf{u}, \mathbf{v})$ . Построить в координатах  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  поверхности постоянных значений общих неортогональных координат:

$$\mathbf{u}(\mathbf{v}, \mathbf{z}) = \text{const}, \mathbf{v}(\mathbf{u}, \mathbf{z}) = \text{const}, \mathbf{z}(\mathbf{u}, \mathbf{v}) = \text{const}.$$

15. а) Составьте в MathCAD программу для вычисления чисел Бернулли, используя выражение для  $\zeta(\mathbf{x})$  – функции Римана с точностью  $\varepsilon = 10^{-6}$ :

Числа Бернулли:

$$B_0 = 1, \quad B_1 = -\frac{1}{2}, \quad B_2 = \frac{1}{6}, \quad B_{2k+1} = 0 \quad \text{для } k = 1, 2, \dots$$

$\zeta$  – функция Римана:  $\zeta(x) = \sum_{k=1}^{\infty} \frac{1}{k^x} = 1 + \frac{1}{2^x} + \frac{1}{3^x} + \dots$

При  $x$  целом и четном  $x = 2n$ , этот ряд связан с числами Бернулли:

$$\zeta(2n) = \frac{(-1)^{n+1} (2\pi)^{2n} B_{2n}}{2(2n)!} \quad \text{при } n = 1, 2, 3, \dots$$

Выведите значения чисел Бернулли для  $n$  из диапазона  $[0; 16]$ . Постройте график  $\zeta(x)$  функции Римана в диапазоне  $x \in [1.2; 6]$ .

б) Запишите в MathCAD выражение для вычисления цилиндрической функции Неймана  $N_n(x)$  с точностью  $\varepsilon = 10^{-6}$ :

$$N_n(x) = \frac{J_n(x) \cos n\pi - J_{-n}(x)}{\sin n\pi}, \quad J_{-n}(x) = (-1)^n J_n(x),$$

$$J_n(x) = x^n \left[ 1 - \frac{x^2}{2(2n+2)} + \frac{x^4}{2 \cdot 4(2n+2)(2n+4)} - \frac{x^6}{2 \cdot 4 \cdot 6(2n+2)(2n+4)(2n+6)} + \dots \right] +$$

$$+ x^{-n} \left[ 1 + \frac{x^2}{2(2n-2)} + \frac{x^4}{2 \cdot 4(2n-2)(2n-4)} + \dots \right].$$

Построить график функции Неймана  $N_n(x)$  в диапазоне в диапазоне  $x \in [-1; 1]$  при различных нецелых значениях параметра  $n$  из диапазона  $[2; 5]$ .

с) Общие ортогональные системы координат  $u, v, z$ :

$$u + iv = f(x + iy), \quad z = z,$$

где  $f(x + iy)$  – аналитическая функция.

Составить в MathCAD программу, преобразования координат точки  $(u, v, z)$  в  $(x, y, z)$ , если

$$u + iv = \ln(x + iy),$$

$$u = \ln\left(\sqrt{x^2 + y^2}\right), \quad x = e^u \cos v, \quad -\infty < u < +\infty,$$

$$v = \operatorname{arctg} \frac{y}{x}, \quad y = e^u \sin v, \quad 0 \leq v < 2\pi.$$

Построить в координатах  $(x, y, z)$  поверхности постоянных значений общих ортогональных координат:

$$u(v, z) = \text{const}, v(u, z) = \text{const}, z(u, v) = \text{const}.$$



### 1.2.2.2 Варианты заданий регрессии экспериментальных данных

16. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1.8; -1.6; -1.4; -1.2; -1; -0.8; -0.6; -0.4; -0.2; 0; 0.2; 0.4; 0.6; 0.8; 1; 1.2; 1.4; 1.6; 1.8; 2; 2.2; 2.4; 2.6

**Y:** 9.9; 6.9; 5.7; 7.5; 7.2; 5.9; 5.3; 3; 4.1; 2.7; 1.7; 1.1; 0.86; 0.21; -0.37; -1.3; -1.6; -2.8; -3.2; -2.6; -3.4; -5; -6.4; -5.8

- Многомерная полиномиальная регрессия по данным:

**X:** 68; 17; 18; 77; 54; 20; 38; 44; 66; 7; 52; 54; 92; 34; 67; 64; 83; 35; 78; 54

**Y:** 60; 62; 6; 44; 10; 84; 75; 80; 91; 51; 21; 15; 61; 86; 90; 24; 86; 79; 66; 7

**Z:** 4.7; 2.8; 0.71; 3; 5; 3.7; 8.9; 8.8; 2.6; 3.1; 6.4; 6; 4.6; 5.5; 8.7; 9.3; 5.5; 5.9; 0.4; 5.6

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

17. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1.8; -1.6; -1.4; -1.2; -1; -0.8; -0.6; -0.4; -0.2; 0; 0.2; 0.4; 0.6; 0.8; 1; 1.2; 1.4

**Y:** -7.1; -5.7; -6.6; -4.2; -5.0; -4.1; -3.5; -2.9; -3.2; -2.3; -1.7; -1.5; -0.9; -0.6; -0.5; 0.1; 0.3; 0.8

- Многомерная полиномиальная регрессия по данным:

**X:** 2; 3; 5; 18; 14; 6; 2; 9; 17; 9; 5; 11; 11; 12; 6; 6; 4; 3; 11; 12; 3; 14; 13

**Y:** 16; 7; 4; 3; 4; 13; 12; 12; 15; 15; 10; 5; 1; 14; 20; 6; 19; 16; 2; 0; 2; 3; 18

**Z:** 5.3; 4.2; 7.7; 9.0; 1.5; 1.2; 5.3; 8.0; 1.3; 1.3; 5.9; 8.6; 1.0; 4.6; 0.4; 2.8; 2.6; 9.2; 7.5; 3.7; 3.7; 5.2; 9.8

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

18. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1.8; -1.6; -1.4; -1.2; -1; -0.8; -0.6; -0.4; -0.2; 0; 0.2; 0.4; 0.6; 0.8; 1; 1.2; 1.4; 1.6; 1.8; 2; 2.2

**Y:** -5.35; -6.76; -6.19; -5.66; -5.58; -4.80; -3.82; -2.59; -2.51; -2.82; -2.07; -2.02; -1.53; -0.89; -0.48; 0.01; 0.32; 0.61; 1.23; 1.47; 1.59; 2.40

- Многомерная полиномиальная регрессия по данным:

**X:** 12; 8; 12; 7; 7; 16; 11; 13; 3; 5; 17; 1; 7; 6; 5; 3; 14; 11; 20

**Y:** 10,00; 10,00; 15,00; 11,00; 2,00; 20,00; 1,00; 13,00; 11,00; 15,00; 13,00; 8,00; 6,00; 5,00; 18,00; 12,00; 11,00; 19,00; 6,00

**Z:** 6,04; 7,15; 5,65; 7,36; 2,42; 7,88; 8,62; 8,97; 1,67; 8,51; 7,09; 7,54; 4,91; 1,87; 0,55; 4,53; 3,43; 6,27; 9,53

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

19. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4; 1,6

**Y:** -4,15; -4,94; -4,37; -4,10; -2,74; -3,26; -3,69; -2,22; -3,21; -2,10; -2,03; -1,61; -1,62; -1,02; -1,09; -0,47; -0,41; -0,03; 0,24

- Многомерная полиномиальная регрессия по данным:

**X:** 10; 11; 10; 16; 4; 15; 9; 2; 20; 14; 1; 7; 13; 17; 0; 6; 9

**Y:** 6; 14; 2; 3; 15; 6; 7; 2; 14; 8; 7; 20; 14; 1; 6; 15; 0

**Z:** 7,7; 2,1; 1,3; 10,0; 3,1; 4,0; 7,1; 5,5; 2,1; 3,1; 0,8; 7,4; 7,7; 4,3; 6,5; 0,4; 6,6

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

20. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2

**Y:** -1,06; -1,27; -0,76; -0,44; -0,43; -0,07; 0,21; 0,38; 0,64; 1,07; 0,84; 1,31; 1,61; 2,04; 2,09; 3,00; 2,76

- Многомерная полиномиальная регрессия по данным:

**X:** 15; 4; 3; 20; 6; 8; 14; 11; 4; 6; 2; 15; 15; 9; 13; 1; 13; 0

**Y:** 16; 19; 4; 20; 7; 1; 8; 3; 14; 6; 10; 18; 2; 14; 3; 5; 2; 14

**Z:** 2,85; 7,43; 5,44; 3,54; 5,57; 1,30; 5,03; 3,18; 2,33; 1,87; 7,79; 6,64; 1,26; 2,31; 2,24; 0,61; 2,53; 4,08

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

21. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4; 1,6; 1,8; 2

**Y:** 6,14; 5,26; 7,46; 6,74; 4,85; 3,39; 4,25; 2,08; 2,05; 1,57; 1,26; 0,31; -0,21; -0,91; -1,01; -2,35; -3,33; -3,14; -3,58; -5,40; -5,76

- Многомерная полиномиальная регрессия по данным:

**X:** 1; 15; 5; 10; 1; 14; 9; 10; 12; 5; 7; 1; 2; 6; 2; 5; 10; 15; 5; 5; 5; 4; 16; 10; 10

**Y:** 4; 15; 2; 15; 17; 3; 17; 1; 1; 17; 8; 8; 2; 7; 20; 15; 3; 12; 10; 1; 18; 0; 13; 11; 8

**Z:** 3,89; 6,51; 6,93; 7,92; 6,30; 1,70; 5,36; 9,29; 8,56; 2,10; 9,75; 5,99; 2,95; 8,32; 5,62; 7,47; 4,44; 4,88; 3,61; 9,58; 5,61; 5,21; 4,51; 0; 0

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.85**.

22. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4; 1,6; 1,8; 2; 2,2; 2,4; 2,6; 2,8; 3

**Y:** -4,34; -5,18; -5,02; -3,70; -3,30; -3,23; -3,14; -4,63; -3,54; -3,87; -2,42; -2,52; -2,9; -2,31; -2,61; -2,56; -1,49; -1,77; -1,17; -1,20; -1,08; -0,77; -0,60; -0,40; -0,20; 0,00

- Многомерная полиномиальная регрессия по данным:

**X:** 8; 16; 6; 6; 16; 6; 18; 3; 8; 4; 12; 8; 14; 9; 8; 16; 14; 20; 15; 8; 17; 9; 0

**Y:** 15; 12; 4; 1; 12; 15; 12; 16; 7; 14; 7; 7; 1; 3; 17; 11; 13; 8; 15; 4; 10; 3; 8

**Z:** 4,40; 3,46; 1,43; 5,74; 2,90; 7,12; 8,45; 5,57; 1,15; 0,09; 3,33; 2,76; 4,81; 3,22; 3,69; 5,75; 9,67; 0,33; 0,63; 0,90; 0,75; 1,22; 1,15

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

23. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4; 1,6; 1,8; 2; 2,2; 2,4; 2,6; 2,8

**Y:** -0,38; -0,14; 0,09; 0,30; 0,70; 0,88; 1,39; 1,92; 2,35; 1,58; 2,13; 2,36; 2,77; 4,14; 3,49; 3,99; 3,72; 3,91; 4,50; 6,20; 4,10; 5,48; 6,10; 6,40; 6,70

- Многомерная полиномиальная регрессия по данным:

**X:** 16; 16; 5; 8; 9; 1; 15; 17; 13; 9; 12; 14; 18; 12; 3; 17; 0

**Y:** 7; 16; 8; 17; 4; 12; 18; 12; 17; 16; 2; 4; 9; 3; 1; 18; 0

**Z:** 1,10; 3,34; 8,97; 3,54; 4,66; 9,77; 5,96; 8,43; 6,64; 9,77; 5,39; 8,11; 7,17; 4,44; 7,15; 1,59; 7,13

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.85**.

24. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4

**Y:** 4,26; 5,84; 5,82; 3,94; 3,32; 4,30; 4,35; 3,19; 2,56; 2,04; 2,60; 1,49; 1,60; 0,82; 0,48; 0,18; -0,15; -0,67

- Многомерная полиномиальная регрессия по данным:

**X:** 15; 12; 17; 1; 9; 5; 12; 10; 16; 4; 9; 11; 7; 1; 13; 15; 16; 0

**Y:** 3; 5; 5; 19; 13; 10; 14; 4; 10; 16; 19; 15; 1; 16; 18; 19; 4; 0

**Z:** 3,27; 2,23; 3,72; 2,89; 4,95; 1,01; 1,35; 4,15; 4,37; 3,43; 1,95; 3,51; 4,25; 3,96; 0,61; 1,85; 2,71; 0,55

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.85**.

25. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** 0; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4; 1,6; 1,8; 2; 2,2; 2,4; 2,6; 2,8; 3; 3,2

**Y:** -5,43; -4,87; -4,75; -3,12; -3,41; -2,90; -3,67; -3,44; -2,62; -2,75; -1,57; -2,18; -1,80; -1,38; -0,88; -0,55; -0,18; 0,10; 0,32; 0,70; 0,72; 1,01; 1,31; 1,80; 1,72; 2,48; 2,80

- Многомерная полиномиальная регрессия по данным:

**X:** 8; 18; 17; 4; 3; 3; 14; 18; 13; 5; 2; 2; 16; 14; 15; 10; 14; 20; 7; 16; 4; 4

**Y:** 13; 11; 2; 16; 2; 13; 13; 13; 8; 7; 6; 13; 9; 12; 2; 18; 9; 6; 1; 18; 16; 15

**Z:** 2,46; 0,43; 2,18; 0,83; 1,58; 0,36; 9,76; 4,68; 6,60; 6,73; 2,46; 2,38; 9,74; 0,27; 3,91; 1,55; 3,33; 8,27; 0,29; 1; 7,82; 9,44

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

26. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,75; -1,5; -1,25; -1; -0,75; -0,5; -0,25; 0; 0,25; 0,5; 0,75; 1; 1,25; 1,5; 1,75

**Y:** 5,12; 5,06; 4,86; 4,24; 4,76; 3,88; 3,24; 3,27; 2,52; 2,32; 2,00; 1,38; 0,99; 1,11; 0,65; 0,23

- Многомерная полиномиальная регрессия по данным:

**X:** 16; 8; 11; 13; 14; 3; 20; 17; 7; 3; 10; 3; 8; 11; 1; 10; 1; 20; 10; 3; 10

**Y:** 16; 17; 9; 9; 15; 15; 9; 15; 13; 18; 8; 1; 10; 14; 16; 17; 12; 1; 8; 8; 11

**Z:** 5,02; 4,78; 6,42; 4,30; 7,80; 3,65; 2,33; 6,54; 3,89; 10; 7,42; 8,35; 4,43; 1,26; 1,87; 7,85; 5,49; 1,69; 1,03; 3,71; 9,17

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

27. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4; 1,6; 1,8; 2; 2,2; 2,4

**Y:** 4,86; 5,80; 4,69; 4,04; 3,82; 2,94; 3,89; 4,17; 3,22; 3,19; 2,81; 2,65; 1,84; 1,48; 1,00; 0,90; 0,64; 0,29; 0,07; -0,24; -0,40; -0,91; -1,10

- Многомерная полиномиальная регрессия по данным:

**X:** 18; 13; 15; 17; 10; 3; 5; 4; 1; 18; 15; 12; 17; 3; 8; 5; 3; 9; 0; 7

**Y:** 16; 14; 16; 1; 15; 17; 16; 14; 19; 1; 13; 9; 3; 8; 5; 13; 18; 16; 14; 20

**Z:** 7,97; 1,86; 5,69; 2,22; 4,38; 6,08; 9,97; 8,22; 3,24; 3,82; 3,05; 6,63; 2,12; 8,48; 3,74; 2,51; 0,42; 4,67; 7,90; 0,78

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

28. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2

**Y:** 6,57; 4,23; 5,78; 5,63; 5,48; 4,01; 2,85; 2,90; 3,74; 3,41; 2,50; 2,08; 2,15; 1,13; 1,30; 0,99; 0,70

- Многомерная полиномиальная регрессия по данным:

**X:** 4; 4; 14; 12; 14; 2; 8; 10; 14; 4; 9; 0; 10; 13; 0; 20; 2; 15; 15; 6

**Y:** 9; 10; 10; 6; 18; 9; 16; 15; 19; 9; 13; 4; 13; 14; 14; 1; 14; 10; 20; 11

**Z:** 2; 0,09; 4,79; 5,68; 0,21; 9,09; 8; 5,12; 2,42; 5,38; 0,53; 9,85; 5,70; 8,11; 8,44; 1,31; 9,31; 8,15; 1,14; 1,02

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

29. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -2; -1,8; -1,6; -1,4; -1,2; -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1

**Y:** -0,11; 0,07; 0,24; 0,35; 0,46; 0,56; 0,84; 1,20; 1,08; 1,67; 1,71; 1,30; 1,88; 1,39; 2,28; 2,30

- Многомерная полиномиальная регрессия по данным:

**X:** 7; 5; 12; 5; 18; 9; 8; 6; 20; 7; 15; 7; 19; 5; 3; 12; 5; 9; 0; 18; 6; 4; 6

**Y:** 9; 4; 1; 3; 7; 12; 18; 13; 7; 14; 16; 6; 14; 20; 5; 10; 10; 10; 16; 5; 0; 15; 13

**Z:** 1,91; 2,35; 4,91; 3,83; 3,12; 1,25; 1,67; 1,25; 3,81; 0,54; 2,01; 2,16; 3,40; 2,20; 4,85; 4,46; 4,53; 1,62; 2,78; 2,79; 1,71; 3,42; 3,79

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

30. Выполните регрессию экспериментальных данных:

- Линейная регрессия:  $y = a x + b$ . Найти значения параметров **a** и **b** по данным:

**X:** -1; -0,8; -0,6; -0,4; -0,2; 0; 0,2; 0,4; 0,6; 0,8; 1; 1,2; 1,4; 1,6; 1,8; 2; 2,2; 2,4; 2,6

**Y:** 4,18; 3,26; 2,85; 3,32; 2,23; 2,23; 1,24; 1,69; 1,24; 0,94; 0,49; 0,16; - 0,11; -0,40; -0,60; -0,94; -1,65; -1,55; -1,90

- Многомерная полиномиальная регрессия по данным:

**X:** 11; 11; 13; 7; 6; 1; 5; 9; 8; 1; 11; 11; 19; 7; 3; 14; 3

**Y:** 12; 19; 13; 19; 16; 5; 15; 19; 9; 15; 13; 6; 14; 11; 3; 10; 11

**Z:** 4,30; 3,80; 0,67; 3,55; 4,98; 1,37; 3,08; 0,75; 5,12; 6; 2,84; 9,67; 5,44; 1,04; 4,69; 0,45; 3,09

Используйте полиномы степени **n = 3**. При построении сглаживающего полинома (**loess**) используйте параметр **span = 0.75**.

### 1.3 Контрольные вопросы

1. В какой версии MathCAD реализована возможность составлять программы?
2. Как в MathCAD организуются циклы (в том числе вложенные) с помощью ранжированных переменных?
3. Как называется специальная панель инструментов для вставки программного кода в документы в MathCAD?
4. Перечислите преимущества программных модулей по сравнению с использованием встроенных функций MathCAD.
5. Как обозначается в MathCAD программный модуль?
6. Перечислите действия, необходимые для создания программного модуля.
7. Как редактировать созданный программный код?
8. Опишите, как работают условные операторы.
9. Опишите, как работают операторы цикла.

## ЛАБОРАТОРНАЯ РОБОТА 2

### РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ И СИСТЕМ

Цель работы – Научиться работать с программой MathCAD. Проводить решение дифференциальных уравнений и систем с использованием программы MathCAD.

#### 2.1 Общие сведения

##### 2.1.1 Решение дифференциального уравнения

Для численного решения одиночного дифференциального уравнения в MathCAD имеется функция `Odesolve`, с помощью которой может быть решена как *задача Коши* (заданы начальные условия) для обыкновенного дифференциального уравнения, так и *краевая задача* (заданы значения функции или ее производных на границе участка числовой оси, на котором ищется решение). Эта функция входит в состав блока решения и является его заключительным ключевым словом.

`Odesolve(x,b,[step])` – Возвращает функцию, которая является решением дифференциального уравнения:  $x$  - переменная интегрирования, действительное число;  $b$  - конечная точка отрезка интегрирования;  $step$  - величина шага по переменной интегрирования (необязательный аргумент). Используется в блоке с оператором `Given`, что показано на рис. 2.1.

`Given`

$$100 \cdot y''(x) + 10 \cdot y'(x) + 101 \cdot y(x) = 50 \cdot \cos\left(\frac{1}{4} \cdot x\right)$$

$$y(0) = 0 \quad y'(0) = 1$$

$$y := \text{odesolve}(x, 150) \quad y(0) = 0 \quad y(50) = 0.476$$

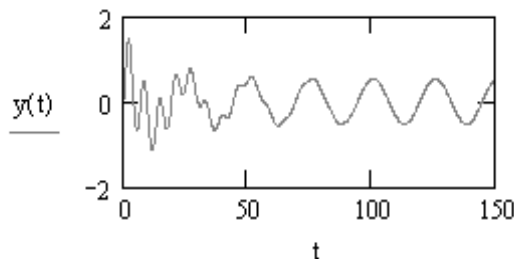


Рисунок 2.1 - Решение дифференциального уравнения.

Замечания:

- Уравнение должно быть линейным относительно старшей производной.
- Число заданных начальных или граничных условий внутри блока должно быть равно порядку уравнения.
- При записи уравнения для обозначения производных функции используйте специальные кнопки с панели Math или ' (штрих) – [Ctrl+F7], для знака равенства = [Ctrl+=] (в том числе и для дополнительных условий).
- Конечная точка должна быть больше начальной.
- Не допускаются начальные и граничные условия смешанного типа ( $f'(a) + f(a)=5$ ).

Искомая функция в блоке должна быть обязательно с аргументом ( $f(x)$ ).

### 2.1.2 Численное решение задачи Коши для дифференциальных уравнений и систем

Для численного решения задачи Коши для дифференциальных уравнений и систем могут быть использованы функции:

- **Rkfixed(y,x1,x2,n,F)** - возвращает матрицу решений системы уравнений методом Рунге-Кутты 4-го порядка при фиксированном шаге по  $x$ .
- **Rkadapt(y,x1,x2,n,F)** - ищет решение с переменным шагом (там, где решение меняется медленнее, шаг увеличивается, а в области быстрого изменения решения шаг функции уменьшается). Возвращается решение с равным шагом. Функция работает быстрее, чем **Rkfixed**.
- **Bulstoer(y,x1,x2,n,F)** - дает более точное решение (методом Bulirsch-Stoer)

Аргументы вышеуказанных функций:

- **y** - вектор начальных условий;
- **x1, x2** - границы интервала для поиска решения;
- **n** - количество точек на интервале;
- **F(x, y)** - вектор-функция первых производных.

При решении дифференциальных уравнений порядка выше первого (или систем уравнений, выше первого порядка) исходное уравнение (систему) необходимо преобразовать к системе дифференциальных уравнений первого порядка (рис. 2.2 - 2.4).

В результате работы указанных функций рассчитывается матрица, количество столбцов которой равно порядку уравнения +1 (или сумме порядков уравнений в системе +1), а количество строк равно параметру **n**. Первый столбец содержит значения независимой переменной, второй – значение функции, третий – для диф. уравнений 2-го порядка – значение



производной искомой функции (если решается система двух уравнений 1-го порядка, то третий столбец будет содержать значения второй функции).

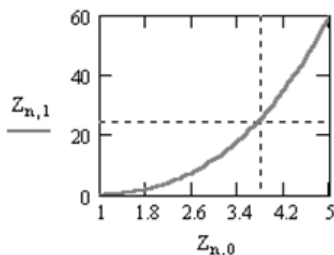
Пример 1. Решить дифференциальное уравнение

$$\frac{d}{dx}y = \frac{y}{x} + x^2$$

при начальном условии  $y(1)=0$  на отрезке  $[1,5]$

$$y_0 := 0 \quad F(x, y) := \frac{y_0}{x} + x^2$$

$$Z := \text{rkfixed}(y, 1, 5, 40, F) \quad n := 0..40$$



	0	1
0	1	0
1	1.1	0.115
2	1.2	0.264
3	1.3	0.448
4	1.4	0.672
5	1.5	0.937
6	1.6	1.248
7	1.7	1.606
8	1.8	2.016
9	1.9	2.479
10	2	3
11	2.1	3.58
12	2.2	4.224
13	2.3	4.933
14	2.4	5.712
15	2.5	6.562

$Z =$

Рисунок 2.2 - Численное решение задачи Коши для дифференциальных уравнений с помощью функции **rkfixed**(y,x1,x2,n,F).

Пример 2. Решить систему диф. уравнений 1-го порядка

$$\frac{d}{dt}x = y - x^2 - x \quad \frac{d}{dt}y = 3 \cdot x - x^2 - y$$

с начальными условиями  $x(0)=0$ ,  $y(0)=1$   
(заменяем  $x$  на  $y_0$ ,  $y$  на  $y_1$ )

$$y := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad F(t, y) := \begin{bmatrix} y_1 - (y_0)^2 - y_0 \\ 3 \cdot y_0 - (y_0)^2 - y_1 \end{bmatrix}$$

$$U := \text{rkfixed}(y, 0, 10, 100, F)$$

Рисунок 2.3 - Численное решение задачи Коши для систем дифференциальных уравнений с помощью функции **Rkfixed**(y, x1, x2, n, F).

Пример 3. Решить диф. уравнение

второго порядка  $\frac{d^2}{dx^2}y - 2\frac{d}{dx}y = e^x + x \cos(x)$

с начальными условиями  $y(0)=0, \frac{d}{dx}y(0) = 1$

Обозначим искомую функцию

$y$  через  $y_0, \frac{d}{dx}y$  через  $y_1$

и запишем наше уравнение в

эквивалентной форме - в виде системы двух уравнений 1-го порядка

$$\begin{bmatrix} \frac{d}{dx}y_0 = y_1 & y_0(0) = 0 \\ \frac{d}{dx}y_1 = 2y_1 - 2y_0 + (e^x + x \cos(x)) & y_1(0) = 1 \end{bmatrix}$$

$$y := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad F(x, y) := \begin{bmatrix} y_1 \\ 2y_1 - 2y_0 + (e^x + x \cos(x)) \end{bmatrix}$$

$Z := \text{rkfixed}(y, 0, \pi, 314, F)$

Пример 4. Решить систему диф. уравнений 2-го порядка

$$\begin{bmatrix} \frac{d^2}{dt^2}u = 2 \cdot v \\ \frac{d^2}{dt^2}v = 4v - 2u \end{bmatrix} \quad \text{с начальными}$$

$$\text{условиями} \begin{pmatrix} u(0) = 1.5 & \frac{d}{dt}u(0) = 1.5 \\ v(0) = 1 & \frac{d}{dt}v(0) = 1 \end{pmatrix}$$

$$y := \begin{pmatrix} 1.5 \\ 1.5 \\ 1 \\ 1 \end{pmatrix} \quad F(t, y) := \begin{bmatrix} y_1 \\ 2y_2 \\ y_3 \\ 4y_2 - 2y_0 \end{bmatrix}$$

$Z := \text{rkfixed}(y, 0, 1, 100, F)$

Рисунок 2.4 - Численное решение систем уравнений с помощью функции `rkfixed(y,x1,x2,n,F)`.

Для выделения решений (функций или их производных) можно воспользоваться стандартным оператором вывода столбцов матрицы **M >**

Если матрица правых частей дифференциальных уравнений почти вырождена, то такие системы называются жесткими. В этом случае решения, возвращаемые функцией **rkfixed**, будет неустойчивым, и для решения таких систем необходимо применять функции **Stiffb**, **Stiff**.

**Stiffb(y,x1,x2,n,F,J)** – ищет решение диф. уравнения или системы дифференциальных уравнений методом **Bulirsch-Stoer**.

**Stiff(y,x1,x2,n,F,J)** – ищет решение диф. уравнения или системы дифференциальных уравнений методом **Rosenbrock**.

Первые пять аргументов такие же, как и при решении хорошо обусловленных систем дифференциальных уравнений.

Дополнительный аргумент - матрица **J** размером  $n(n+1)$ , первый столбец которой содержит частные производные  $dF/dx$ , остальные столбцы и строки представляют собой матрицу Якоби  $dF/dy$  (рис. 2.5).

Для отыскания решения системы диф. уравнений только в конечной точке используются функции **Bulstoer**, **Rkadapt**, **Stiffb**, **Stiff** (начинаются с прописной буквы).

$$J(x,y) = \begin{bmatrix} \frac{\partial F_0}{\partial x} & \frac{\partial F_0}{\partial y_0} & \dots & \frac{\partial F_0}{\partial y_n} \\ \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y_0} & \dots & \frac{\partial F_1}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x} & \frac{\partial F_n}{\partial y_0} & \dots & \frac{\partial F_n}{\partial y_n} \end{bmatrix}$$

Рисунок 2.5 - Матрица J размером  $n(n+1)$ , первый столбец которой содержит частные производные  $dF/dx$ , остальные столбцы и строки представляют собой матрицу Якоби  $dF/dy$

Набор параметров для этих функций :

- Bulstoer(y,x1,x2,acc,F,kmax,save);
- Rkadapt(y,x1,x2,acc,F,kmax,save);
- Stiffb(y,x1,x2,acc,F,J,kmax,save);
- Stiffr(y,x1,x2,acc,F,J,kmax,save).

Первые три параметра и пятый (F) этих функций те же, что и для функции Rkadapt. Дополнительные параметры:

- acc - параметр, контролирующий точность решения (рекомендуется  $acc = 0.001$ );
- kmax - максимальное число промежуточных точек, в которых ищется решение;
- save - минимально допустимый интервал между точками, в которых ищется решение.

### 2.1.3 Решение краевых задач для обыкновенных дифференциальных уравнений

Если для дифференциального уравнения  $n$ -го порядка  $k$  граничных условий заданы в начальной точке  $x_1$ , а  $(n - k)$  граничных условий - в конечной точке  $x_2$ , то такая задача называется краевой. В MathCAD реализованы две функции, позволяющие численно найти недостающие условия в точках  $x_1$  и  $x_2$ .

#### 2.1.3.1 Двухточечная краевая задача

Задача решается в два этапа. Сначала с помощью функции **sbval** находятся недостающие начальные значения, а затем применяется одна из выше описанных функций для решения стандартной задачи Коши на отрезке.

**sbval(v,x1,x2,F,load,score)** - ищет недостающие начальные условия в точке **x1**;

- **v** - вектор начальных приближений для искомых начальных значений в точке **x1**;

- $x1, x2$  - граничные точки интервала;
- $F(x, y)$  - вектор-столбец из  $n$  элементов, содержит правые части дифференциальных уравнений;
- $load(x1, v)$  - вектор-столбец из  $n$  элементов, содержит начальные значения в точке  $x1$ ; некоторые из значений – константы, другие неизвестны и будут найдены в процессе решения;
- $score(x2, y)$  - вектор-столбец размерности вектора  $v$ , содержащий разность между начальным условием в точке  $x2$  и значением искомого решения в этой точке.

Пример решения приведен на рис. 2.6.



Рисунок 2.6 - Решение дифференциального уравнения (двухточечной краевой задачи) третьего порядка

### 2.1.3.2 Краевая задача с условиями внутри интервала

На первом этапе используется функция

**balfit**(V1, V2, x1, x2, xf, F, load1, load2, score) – ищет недостающие начальные условия в точках  $x1$  и  $x2$ , сшивая решения, выходящие из этих точек, в точке  $xf$ ;

- $V1, V2$  – вектора начальных приближений для искомых начальных значений в точках  $x1$  и  $x2$ ;
- $x1, x2$  – граничные точки интервала;
- $load1(x1, V1)$  – вектор-столбец из  $n$  элементов, содержит начальные значения в точке  $x1$ ; некоторые из значений – константы, другие неизвестны и будут найдены в процессе решения;
- $load2(x2, V2)$  – вектор-столбец из  $n$  элементов, содержит начальные значения в точке  $x2$ ; некоторые из значений – константы, другие неизвестны и будут найдены в процессе решения;

- $\text{score}(xf, y)$  – вектор-столбец размерности  $n$ , содержащий разность между решениями, начинающимися в точках  $x_1$  и  $x_2$ , в точке  $xf$ .

Пример приведен на рис. 2.7.

Требуется решить дифференциальное уравнение  $y''=y$  при граничных условиях  $y(-1)=1$  и  $y(3)=2$ , сшивая решения в т.  $x=0$

$$F(x, y) := \begin{pmatrix} y_1 \\ y_0 \end{pmatrix} \quad \begin{matrix} x_1 := -1 & xf := 0 & v1_0 := 1 \\ x_2 := 3 & & v2_0 := 1 \end{matrix}$$

$$\text{load1}(x_1, v_1) := \begin{pmatrix} 1 \\ v1_0 \end{pmatrix} \quad \text{load2}(x_2, v_2) := \begin{pmatrix} 2 \\ v2_0 \end{pmatrix}$$

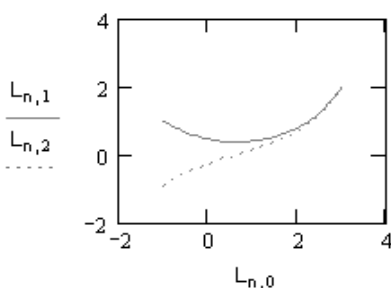
$$\text{score}(xf, y) := y \quad S1 := \text{bvalfit}(v1, v2, x_1, x_2, xf, F, \text{load1}, \text{load2}, \text{score})$$

$S1 = (-0.927 \quad 1.965)$  Недостающие значения производных в точках  $x = -1$  и  $x = 3$ , соответственно

$$y := \begin{pmatrix} 1 \\ -0.927 \end{pmatrix} \quad \text{Составляем вектор начальных условий}$$

$$L := \text{rkfixed}(y, x_1, x_2, 40, F)$$

$$n := 0 \dots 40$$



$L =$

	0	1	2
0	-1	1	-0.927
1	-0.9	0.912	-0.831
2	-0.8	0.833	-0.744
3	-0.7	0.763	-0.665
4	-0.6	0.7	-0.591
5	-0.5	0.645	-0.524
6	-0.4	0.595	-0.462
7	-0.3	0.552	-0.405
8	-0.2	0.514	-0.352
9	-0.1	0.482	-0.302
10	0	0.454	-0.255
11	0.1	0.43	-0.211
12	0.2	0.411	-0.169
13	0.3	0.397	-0.129
14	0.4	0.386	-0.09
15	0.5	0.379	-0.051

Рисунок 2.7 - Решение краевой задачи с условиями внутри интервала

## 2.1.4 Уравнения в частных производных

Дифференциальные уравнения в частных производных требуют нахождения функции не одной, а нескольких переменных. Эти уравнения включают в себя производные по различным переменным (частные производные). Уравнениями в частных производных описывается множество разнообразных физических явлений, и с их помощью можно с успехом моделировать самые сложные явления и процессы (диффузия, гидродинамика, квантовая механика, экология и т. д.). MathCAD имеет ограниченные возможности по отношению к уравнениям в частных производных. С помощью встроенных функций можно решать лишь некоторые из частных случаев.

### 2.1.4.1 Решение уравнения эллиптического типа

Двумерное уравнение Пуассона – пример уравнения в частных производных эллиптического типа, включающее в себя вторые производные функции  $T(x, y)$  по двум пространственным переменным:

$$T_{xx} + T_{yy} = f(x, y)$$

Требуется решить дифференциальное уравнение  $y''=y$  при граничных условиях  $y(-1)=1$  и  $y(3)=2$ , сшивая решения в т.  $x=0$

```
x1 := -1    xf := 0    v10 := 1
F(x, y) :=  $\begin{pmatrix} y_1 \\ y_0 \end{pmatrix}$     x2 := 3    v20 := 1

load1(x1, v1) :=  $\begin{pmatrix} 1 \\ v1_0 \end{pmatrix}$     load2(x2, v2) :=  $\begin{pmatrix} 2 \\ v2_0 \end{pmatrix}$ 

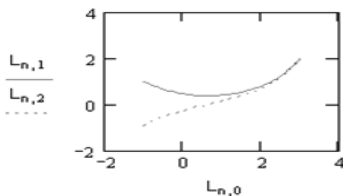
score(xf, y) := y    S1 := bvalfit(v1, v2, x1, x2, xf, F, load1, load2, score)
```

S1 = (-0.927 1.965) Недостаточные значения производных в точках  $x=-1$  и  $x=3$ , соответственно

$y := \begin{pmatrix} 1 \\ -0.927 \end{pmatrix}$  Составляем вектор начальных условий

L := rkfixed(y, x1, x2, 40, F)

n := 0..40



L =

	0	1	2
0	-1	1	-0.927
1	-0.9	0.912	-0.831
2	-0.8	0.833	-0.744
3	-0.7	0.763	-0.665
4	-0.6	0.7	-0.591
5	-0.5	0.645	-0.524
6	-0.4	0.595	-0.462
7	-0.3	0.552	-0.405
8	-0.2	0.514	-0.352
9	-0.1	0.482	-0.302
10	0	0.454	-0.255
11	0.1	0.43	-0.211
12	0.2	0.411	-0.169
13	0.3	0.397	-0.129
14	0.4	0.386	-0.09
15	0.5	0.379	-0.051

Рисунок 2.8 - Решение краевой задачи с условиями внутри интервала

Уравнение Пуассона (Лапласа, если  $f(x, y) \equiv 0$ ) описывает, например, распределение электростатического поля  $E(x, y)$  в двумерной области с плотностью заряда  $q(x, y)$ , или стационарное распределение температуры  $T(x, y)$  на плоскости, в которой имеются источники (или поглотители) тепла с интенсивностью  $f(x, y)$ . Именно в последней физической интерпретации будет далее рассматриваться уравнение Пуассона.

Корректная постановка краевой задачи для уравнения Пуассона требует задания граничных условий. В MathCAD решение ищется на плоской квадратной области, состоящей из  $(m + 1) \times (m + 1)$  точек. Поэтому граничные условия должны быть определены пользователем для всех четырех сторон упомянутого квадрата. Самый простой вариант – это нулевые граничные условия, т. е. постоянная температура по всему периметру расчетной области. В таком случае можно использовать встроенную функцию **multigrid**.

- **multigrid(F, ncycle)** – матрица решения уравнения Пуассона размера  $(M + 1) \times (M + 1)$  на квадратной области с нулевыми граничными условиями;

- **F** – матрица размера  $(M + 1) \times (M + 1)$ , задающая правую часть уравнения Пуассона;

- **ncycle** – параметр численного алгоритма (количество циклов в пределах каждой итерации).

Сторона квадрата расчетной области должна состоять из  $M = 2^n$  точек, где  $n$  – целое число. Параметр численного метода **ncycle** в большинстве случаев достаточно взять равным 2. Листинг 2.1 содержит пример использования функции **multigrid** для расчета краевой задачи на области  $33 \times 33$  точки и точечным источником тепла в месте, задаваемом координатами (15, 20) внутри этой области.

*Листинг 2.1. Решение уравнения Пуассона с нулевыми граничными условиями*

```
M := 32
fM,M := 0
f15,20 := 104
b := multigrid(-f, 2)
```

В первой строке листинга задается значение  $M = 32$ , в двух следующих строках создается матрица правой части уравнения Пуассона, состоящая из всех нулевых элементов, за исключением одного, задающего расположение источника. В последней строке матрице **b** присваивается результат действия функции **multigrid**. Обратите внимание, первый ее

аргумент сопровождается знаком "минус", что соответствует записи правой части уравнения Пуассона. Графики решения показаны на рис. 2.9 в виде трехмерной поверхности (а) и линий уровня (b), соответственно.

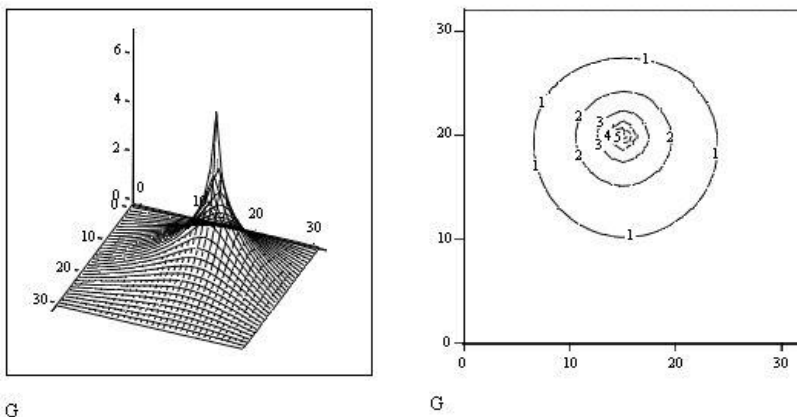


Рисунок 2.9 - График поверхности решения  $G \equiv b$  уравнения Пуассона (а) и график линий уровня решения уравнения Пуассона (b)

В более сложных случаях, например, для решения краевой задачи с ненулевыми условиями на границах, следует использовать другую встроенную функцию **relax**, имеющуюся в MathCAD.

**relax(a, b, c, d, e, F, v, rjac)** – матрица решения дифференциального уравнения в частных производных на квадратной области, полученного с помощью алгоритма релаксации для метода сеток;

- **a, b, c, d, e** – квадратные матрицы коэффициентов разностной схемы, аппроксимирующей дифференциальное уравнение;

- **F** – квадратная матрица, задающая правую часть дифференциального уравнения;

- **v** – квадратная матрица граничных условий и начального приближения к решению;

- **rjac** - параметр численного алгоритма (спектральный радиус итераций Якоби).

Параметр численного алгоритма характеризует скорость сходимости итераций. Он должен быть числом от 0 до 1. В матрице граничных условий **v** необходимо задать только граничные элементы, исходя из значения краевых условий по периметру расчетной области. Прочие (внутренние) элементы этой матрицы служат для задания начального приближения к решению. Суть алгоритма релаксации сводится к тому, что в ходе итераций происходит проверка уравнений и соответствующая коррекция значений искомой функции в каждой точке. Если начальное приближение



выбрано удачно, то можно надеяться, что алгоритм сойдется ("срелаксирует") к правильному решению.

Все матрицы, задающие как коэффициенты разностной схемы **a**, **b**, **c**, **d**, **e**, граничные условия **v**, так и само решение **F**, должны иметь одинаковый размер  $(m + 1) \times (m + 1)$ , соответствующий размеру расчетной области. При этом целое число **M** обязательно должно быть степенью двойки:  $M = 2^n$ . Решение уравнения Пуассона с тремя источниками разной интенсивности при помощи функции **relax** приведено в листинге 2.2.

*Листинг 2.2. Решение уравнения Пуассона с помощью функции **relax**.*

```

M := 32                fM,M := 0

f15,20 := 10          f25,10 := 5          f10,10 := -5
i := 0..M              k := 0..M
ai,k := 1             vi,k := 0

b := a                h := a                d := a                n := -4a
o := relax(a,b,h,d,n,-f,v,.95)

```

Первые три строки имеют тот же смысл, что и в предыдущем листинге. Только вместо одного источника тепла взято их другое распределение – один сильный источник, один более слабый и один сток тепла. В следующих шести строках задаются коэффициенты разностной схемы. В предпоследней строке задана матрица нулевых граничных условий и нулевых начальных приближений, а в последней матрице **G** присваивается результат действия функции **relax**. График полученного решения в виде линий уровня показан на рис. 2.10.

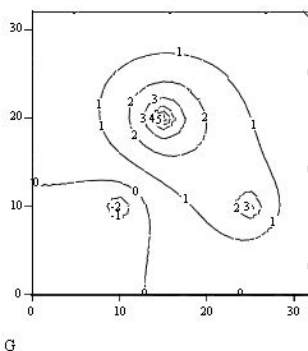


Рисунок 2.10 - Решение уравнения Пуассона с помощью функции **relax** (листинг 2.2)

#### 2.1.4.2 Решение уравнения параболического типа

Уравнение параболического типа содержит первую производную по времени **t** и вторую по пространственной координате **x**.

$$\frac{\partial u(x,t)}{\partial t} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} - v \frac{\partial u(x,t)}{\partial x}$$

Это уравнение описывает динамику температуры **T(x, t)** в присутствии источников тепла **<j>(x, T, t)** при наличии конвекции (член с **u<sub>x</sub>**), например, при нагреве металлического стержня.

Для решения такого уравнения на интервале **[0, L]** по оси **x** и **[0, T]** по оси **t**, используйте блок **Given** с функцией

$$u := \text{Pdesolve}\left[u, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix}, \text{spacepts}, \text{timepts}\right],$$

где **u** – искомая функция;

**x** – пространственная координата;

**t** – временная координата;

**spacepts** – количество точек решения вдоль оси **x**;

**timepts** – количество точек решения на временной оси.

Используйте литеральную систему обозначений нижнего индекса (наберите точку после имени функции, чтобы создать нижние индексы), чтобы записать частные производные в PDE (Partial Differential Equation – уравнение в частных производных), начальных условиях, и граничных условиях. Все уравнения определены с Булевыми знаками равенства – [Ctrl] =.

Given

$$u_t(x,t) = \frac{1}{4} \cdot u_{xxx}(x,t) - \frac{1}{2} \cdot u_x(x,t)$$

Начальные условия:

$$u(x,0) = \sin\left(\pi \cdot \frac{x}{L}\right) + \frac{1}{2} \cdot \sin\left(3 \cdot \pi \cdot \frac{x}{L}\right)$$

Граничные условия Дирихле:

$$u(0,t) = 0 \qquad u(L,t) = 0$$

Решение для **u(x, t)** в диапазоне (координата **x**) от 0 до L и от 0 до T по времени **t** дается соотношением для **u**.

Функция **u** может теперь быть оценена при любом значении **x** и времени **t** в указанных диапазонах. Все обращения к функции и ее производным должны определяться обеими переменными, то есть **u<sub>x</sub>(x, 0)**, а не **u<sub>x</sub>**.

$$u(0.1,0.2) = 0.08 \quad x := 0,0.2..L \quad t := 0..T$$

$$IC(x) := \sin\left(\pi \cdot \frac{x}{L}\right) + \frac{1}{2} \cdot \sin\left(3 \cdot \pi \cdot \frac{x}{L}\right)$$

Выведите на график решение и начальные условия в некоторый момент времени. Переменная **FRAME** была выбрана так, чтобы решение,  $u$ , показанное на рис 2.11, можно было анимировать по времени.

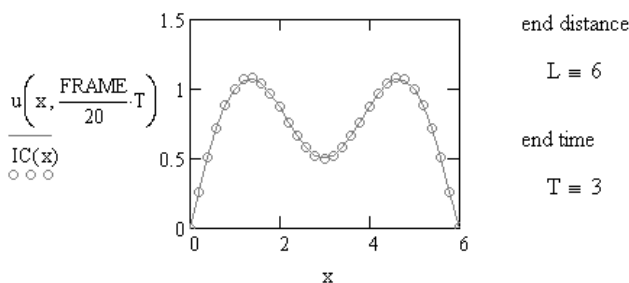
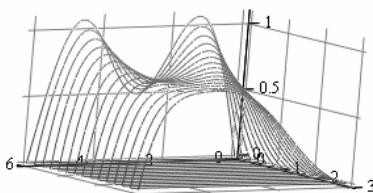


Рисунок 2.11 - Решение уравнения параболического типа с помощью функции **Pdesolve**

Чтобы просмотреть анимацию, выберите **Animation > Record** меню **Tools**, переместите блок выбора вокруг графика, и отрегулируйте число **FRAME** в пределах 20. Затем щелкните кнопку **Animate**.

Можно также изучать полное решение, построив трехмерный график поверхности решения, показанный на рис. 2.12.

$A1 := \text{CreateMesh}(u, 0, L, 0, T, 10 \cdot L, 5 \cdot T)$



A1

Рисунок 2.12 - Построение поверхности полученного решения уравнения параболического типа с помощью функции **Pdesolve**

#### 2.1.4.3 Решение волнового уравнения

##### Использование **PDE solve block**.

Для решения одномерного волнового уравнения

$$\frac{\partial}{\partial t} v(x, t) = a^2 \frac{d^2}{dx^2} w(x, t)$$

используйте ограничение

$$\frac{\partial}{\partial t} w(x, t) = v(x, t)$$

чтобы свести уравнение к системе двух уравнений в частных производных.

Установим **PDE solve block**,

Given

$$v_t(x, t) = a^2 \cdot w_{xx}(x, t) \quad w_t(x, t) = v(x, t)$$

с граничными и начальными условиями:

$$w(x, 0) = \sin\left(\frac{\pi \cdot x}{L}\right) \quad v(x, 0) = 0$$

$$w(0, t) = 0 \quad w(L, t) = 0$$

$$\begin{pmatrix} w \\ v \end{pmatrix} := \text{Pdesolve} \left[ \begin{pmatrix} w \\ v \end{pmatrix}, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix} \right] \quad \begin{array}{l} a \equiv 3 \\ L \equiv 2 \cdot \pi \\ T \equiv 2 \cdot \pi \end{array}$$

Имеется единственное решение на границе, показанное на рис. 2.13.

Можно построить сетку и для нее вычислить решение на плоскости (x, t), как показано на рис. 2.13.

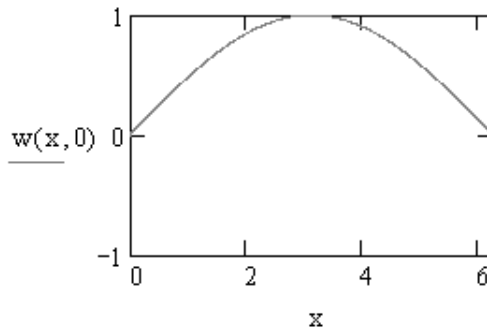
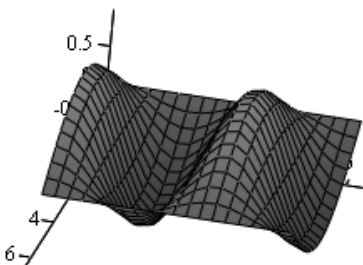


Рисунок 2.13 - Решение волнового уравнения на границе

Можно построить сетку и для нее вычислить решение на плоскости  $(x, t)$ , как показано на рис. 2.14

```
M := CreateMesh(w, 0, L, 0, T)
```



M

Рисунок 2.14 - Решение волнового уравнения на границе.  
Трехмерное изображение поверхности

### Использование функции numol.

Функция **numol(x\_endpts, xpts, t\_endpts, tpts, num\_pde, num\_pae, pde\_func, pinit, bc\_func)** возвращает  $xpts \times tpts$  матрицу, содержащую решение одномерного уравнения в частных производных находящегося в **pde\_func**. Каждая колонка представляет одномерное решение 1-D в один и тот же момент времени. Для системы уравнений, решение для каждой функции добавлено в конец матрицы в виде отдельной строки, так, что матрица всегда имеет **xpts** строк, и **tpts \* (num\_pde + num\_pae)** колонок. Решение найдено численным методом. Пример использования функции **numol** приведен на рис. 2.15.

Аргументы:

- **X\_endpts, t\_endpts** – векторы-колонки с двумя элементами, которые определяют реальные границы областей интегрирования.

- **Xpts, tpts** – количество точек в областях интегрирования для получения приближенного решения.

- **Num\_pde, num\_pae** – количество уравнений в частных производных (УЧП) и алгебраических уравнений (АУ), соответственно. **Num\_pde** должен быть, по крайней мере 1, **num\_pae** может быть 0 или больше.

- **Pde\_func** - векторная функция **x, t, u, u<sub>x</sub>**, и **u<sub>xx</sub>** размером **(num\_pde + num\_pae)**. Она содержит правые стороны УЧП/АУ, и предполагает, что левые стороны – все равны **u**. Решение **u**, предполагается векторной

функцией. Если Вы работаете с системой УЧП, каждый ряд в **Pde\_func** использует векторные нижние индексы: индексы **u**, например, **u[0]** относится к первой функции в системе, а **u.x[1]** относится к первой производной второй функции в системе.

•**Pinit** – векторная функция **x** длины (**num\_pde** + **num\_pae**) содержащая начальные условия для каждой функции в системе.

•**Bc\_func** – **num\_pde** × **3** матрица, содержащая строки в форме:

( Bc_left (t)	Bc_right (t)	"D") для Dirichlet граничных условий, или
( Bc_left (t)	Bc_right (t)	"N") для Neumann граничных условий.

$$\text{num\_pde} := 2 \quad \text{num\_pae} := 0 \quad a \equiv 3 \quad L \equiv 2 \cdot \pi \quad T \equiv 2 \cdot \pi$$

$$\text{rhs}\left(x,t,u,u_x,u_{xx}\right) := \begin{pmatrix} u_1 \\ a^2 \cdot u_{xx} \end{pmatrix} \quad \text{init}(x) := \begin{pmatrix} \sin\left(\frac{\pi \cdot x}{L}\right) \\ 0 \end{pmatrix}$$

$$\text{bc\_func}(t) := \begin{pmatrix} \text{init}(0)_0 \; \text{init}(L)_0 \; \text{"D"} \\ \text{"NA"} \; \text{"NA"} \; \text{"D"} \end{pmatrix}$$

$$\text{sol} := \text{numol}\left[\begin{pmatrix} 0 \\ L \end{pmatrix}, 30, \begin{pmatrix} 0 \\ T \end{pmatrix}, 20, \text{num\_pde}, \text{num\_pae}, \text{rhs}, \text{init}, \text{bc\_func}\right]$$

$$i := 0..30 \qquad x_1 := \frac{i \cdot L}{30} \qquad t0 := 1$$

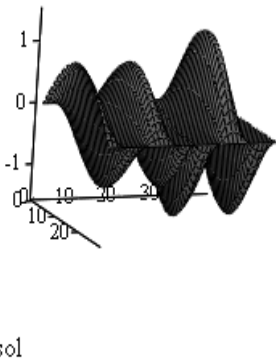
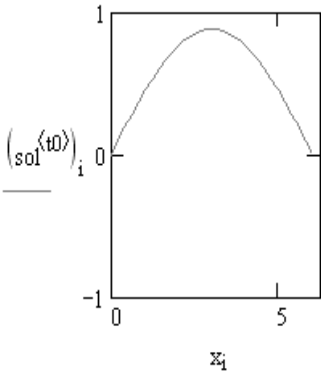


Рисунок 2.15 - Применение функции **numol** для решения волнового уравнения

## 2.2 Порядок выполнения работы

### 2.2.1 Задания для выполнения работы

1. Решите дифференциальные уравнения первого порядка.
2. Решите дифференциальные уравнения второго или высшего порядка.
3. Решите дифференциальные уравнения в частных производных.

### 2.2.2 Варианты заданий

2.2.2.1 Решить дифференциальные уравнения первого порядка  
Решить дифференциальные уравнения первого порядка (задача Коши) в системе MathCAD с помощью блока **Given ... odesolve**. Построить график полученного решения.

1. Решить дифференциальное уравнение первого порядка  $y' \sin x = y \ln y$  на интервале  $\left[\frac{\pi}{2}; 2.5\right]$  при начальном условии  $y\left(\frac{\pi}{2}\right) = e$ .
2. Решить дифференциальное уравнение первого порядка  $y' + 2y = 4x$  на интервале  $[0; 5]$  при начальном условии  $y(0) = 5$ .
3. Решить дифференциальное уравнение первого порядка  $y' = xe^{-x^2} - 2xy$  на интервале  $[0, 3]$  при начальном условии  $y(0) = 0$ .
4. Решить дифференциальное уравнение первого порядка  $y' + y = \cos x$  на интервале  $[0; 50]$  при начальном условии  $y(0) = 1$ .
5. Решить дифференциальное уравнение первого порядка  $(1+x^2)y' - 2xy = (1+x^2)^2$  на интервале  $[0, 2]$  при начальном условии  $y(0) = -2$ .
6. Решить дифференциальное уравнение первого порядка  $y' = e^{mx} - ay$  при  $m = 0.1$  и  $a = 1.5$  на интервале  $[0, 2]$  при начальном условии  $y(0) = -1$ .
7. Решить дифференциальное уравнение первого порядка  $(2x - y^2)y' = 1$  на интервале  $[0, 10]$  при начальном условии  $y(0) = -1$ .
8. Решить дифференциальное уравнение первого порядка  $y' = \frac{y}{2y \ln y + y - x}$  на интервале  $[0, 10]$  при начальном условии  $y(0) = 0.5$ .
9. Решить дифференциальное уравнение первого порядка  $y' + y \operatorname{tg} x = \frac{1}{\cos x}$  на интервале  $[0, 10]$  при начальном условии  $y(0) = 2$ .
10. Решить дифференциальное уравнение первого порядка  $xy' + y = e^x$  на интервале  $[-2, -0.5]$  при начальном условии  $y(-2) = -1$ .
11. Решить дифференциальное уравнение первого порядка  $y' = 1 + \frac{y}{x(x+1)}$  на интервале  $[1, 10]$  при начальном условии  $y(1) = 0$ .

12. Решить дифференциальное уравнение первого порядка  $x^2 y' = x^2 + xy + y^2$  на интервале  $[1, 2]$  при начальном условии  $y(1) = 0$ .

13. Решить дифференциальное уравнение первого порядка  $x^2 y' = 2xy - 3$  на интервале  $[1, 10]$  при начальном условии  $y(1) = 0$ .

14. Решить дифференциальное уравнение первого порядка  $xy' = y + 1$  на интервале  $[1, 10]$  при начальном условии  $y(1) = 0$ .

Решить дифференциальное уравнение первого порядка  $xy(1+x^2)y' = y^2 + 1$  на интервале  $[1, 10]$  при начальном условии  $y(1) = 2$ .

2.2.2.2 Решить дифференциальные уравнения второго или высшего порядка

Решить дифференциальные уравнения второго или высшего порядка (задача Коши) в системе MathCAD с помощью блока **Given ... odesolve**. При решении необходимо использовать все начальные условия для функций и их производных. Построить график полученного решения.

1. Решить дифференциальное уравнение третьего порядка на интервале  $[0.5, 5]$   $y''' = \frac{1}{x}$  при начальных условиях  $y(0.5) = 0.5$ ;  $y'(0.5) = -0.2$ ;  $y''(0.5) = -0.1$ .

2. Решить дифференциальное уравнение второго порядка  $(x^2 + 1)y'' = 2xy'$  на интервале  $[1, 3]$  при начальных условиях  $y(0) = 1$ ;  $y'(0) = 3$ .

3. Решить дифференциальное уравнение третьего порядка  $y''' = \cos 2x$  на интервале  $[0, 10]$  при начальных условиях  $y(0) = 0$ ;  $y'(0) = 0$ ;  $y''(0) = 0$ .

4. Решить дифференциальное уравнение второго порядка на интервале  $[0, 3]$   $y'' = e^{-\frac{x}{2}}$  при начальных условиях  $y(0) = 0$ ;  $y'(0) = -0.5$ .

5. Решить дифференциальное уравнение третьего порядка на интервале  $[0.1, 10]$   $x^2 y''' = y''^2$  при начальных условиях  $y(0.1) = 0$ ;  $y'(0.1) = 0.5$ ;  $y''(0.1) = -0.1$ .

6. Решить дифференциальное уравнение второго порядка  $xy'' = y' - xy'^2$  на интервале  $[2, 10]$  при начальных условиях  $y(2) = 2$ ;  $y'(2) = 1$ .

7. Решить дифференциальное уравнение третьего порядка  $y''' = y''^3$  на интервале  $[0, 2]$  при начальных условиях  $y(0) = -1$ ;  $y'(0) = -0.5$ ;  $y''(0) = 0.3$ .



8. Решить дифференциальное уравнение второго порядка  $y'' = \frac{y'}{x} + \frac{x^2}{y'}$  на интервале  $[2, 5]$  при начальных условиях  $y(2) = 0$ ;  $y'(2) = 4$ .

9. Решить дифференциальное уравнение третьего порядка  $y'y''' = 3y''^2$  на интервале  $[0, 10]$  при начальных условиях  $y(0) = 1$ ;  $y'(0) = -0.1$ ;  $y''(0) = 0.3$ .

10. Решить дифференциальное уравнение второго порядка на интервале  $[-2, 5]$   $2y'' = 3y^2$  при начальных условиях  $y(-2) = 1$ ;  $y'(-2) = -1$ .

11. Решить дифференциальное уравнение третьего порядка  $yy''' = y'y''$  на интервале  $[0, 2]$  при начальных условиях  $y(0) = 1$ ;  $y'(0) = 0.1$ ;  $y''(0) = 0.3$ .

12. Решить дифференциальное уравнение второго порядка  $yy'' = y'^2 - y'^3$  на интервале  $[1, 5]$  при начальных условиях  $y(1) = 1$ ;  $y'(1) = -1$ .

13. Решить дифференциальное уравнение третьего порядка  $y'y''' = y''^2 - \left(\frac{y'}{x}\right)^2$  на интервале  $[0.5, 2]$  при начальных условиях  $y(0.5) = 0$ ;  $y'(0.5) = 0.5$ ;  $y''(0.5) = -1.5$ .

14. Решить дифференциальное уравнение второго порядка  $y^3 y'' = -1$  на интервале  $[1, 2]$  при начальных условиях  $y(1) = 1$ ;  $y'(1) = 0$ .

15. Решить дифференциальное уравнение третьего порядка  $(1 + y'^2) y''' = 3y'y''^2$  на интервале  $[0, 5]$  при начальных условиях  $y(0) = 0$ ;  $y'(0) = 1$ ;  $y''(0) = -0.7$ .

2.2.2.3 Решить дифференциальные уравнения в частных производных

В задании необходимо решить дифференциальные уравнения в частных производных: уравнение Пуассона с нулевыми граничными условиями (используйте функцию **multigrid**); уравнение Пуассона с ненулевыми граничными условиями (используйте функцию **relax**); уравнение параболического типа (используйте блок **Given ... Pdesolve**); одномерное волновое уравнение (используйте блок **Given ... Pdesolve** или функцию **numol**). Для каждого решения построить график начальных условий, двумерный график поверхности решения и график линий уровня.

1. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с ненулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 32$ ), если в точках  $(10, 10)$ ,  $(25, 15)$  и  $(5, 30)$  матрицы правой части уравнения заданы значения  $F_{10, 10} = 1000$ ,  $F_{25, 15} = 500$  и  $F_{5, 30} = -1000$ , параметры уравнения

$b_{i,k} = c_{i,k} = d_{i,k} = a_{i,k} = 1$ , а матрица граничных условий имеет вид  $\mathbf{v}_{i,k} = \mathbf{i} \times (1 - \mathbf{k})$ , а  $\mathbf{i}, \mathbf{k} = 0, 1, \dots, \mathbf{M}$ .

- Уравнение параболического типа  $\mathbf{u}_t = \mathbf{u}_{xx} - 2\mathbf{u}_x$  с граничными условиями Дирихле  $\mathbf{u}(0, t) = 1$ ;  $\mathbf{u}(\mathbf{L}, t) = 0$  и начальными условиями

$u(x, 0) = \cos\left(\pi \frac{x}{L}\right) + \frac{1}{2} \sin\left(3\pi \frac{x}{L}\right)$  при  $\mathbf{L} = 6$  и для конечной точки по времени  $\mathbf{T} = 3$ . Использовать количество точек для решения по координате –  $\mathbf{N}_x = 30$ , и по времени –  $\mathbf{N}_t = 60$ .

- Одномерное волновое уравнение (использовать PDE solve block)  $\mathbf{v}_t(\mathbf{x}, t) = \mathbf{a}^2 \mathbf{w}_{xx}(\mathbf{x}, t)$  с ограничением  $\mathbf{w}_t(\mathbf{x}, t) = \mathbf{v}(\mathbf{x}, t)$  при  $\mathbf{a} = 2$  в диапазонах по координате  $[0, 2\pi]$  и по времени  $[0, 2\pi]$  с начальными условиями  $\mathbf{w}(\mathbf{x}, 0) = \cos(\pi x/L)$  и  $\mathbf{v}(\mathbf{x}, 0) = \mathbf{x}$  и граничными условиями Дирихле  $\mathbf{w}(0, t) = 0$  и  $\mathbf{w}(\mathbf{L}, t) = t$ .

2. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(\mathbf{M} + 1) \times (\mathbf{M} + 1)$  точек ( $\mathbf{M} = 32$ ), если в точках матрицы правой части уравнения заданы значения  $\mathbf{F}_{i,i} = 100 \times i$ .

- Уравнение параболического типа  $\mathbf{u}_t = \mathbf{u}_{xx} - \frac{1}{2}\mathbf{u}_x$  с граничными условиями Дирихле  $\mathbf{u}(0, t) = t / 2$ ;  $\mathbf{u}(\mathbf{L}, t) = 2t$  и начальными условиями

$u(x, 0) = -\sin\left(\pi \frac{x}{L}\right) + x$  при  $\mathbf{L} = 6$  и для конечной точки по времени  $\mathbf{T} = 3$ . Использовать количество точек для решения по координате –  $\mathbf{N}_x = 30$ , и по времени –  $\mathbf{N}_t = 60$ .

- Одномерное волновое уравнение (использовать функцию **numol**)  $\mathbf{v}_t(\mathbf{x}, t) = \mathbf{a}^2 \mathbf{v}_{xx}(\mathbf{x}, t)$  при  $\mathbf{a} = 1/9$  в диапазонах по координате  $[\mathbf{x}_0 = 0, \mathbf{x}_{end} = 3/2]$  и по времени  $[\mathbf{t}_0 = 0, \mathbf{t}_{end} = 3/2]$ . Количество дифференциальных уравнений **num\_pde** = 2; количество алгебраических уравнений **num\_pae**

= 0. Правая часть функции **pde\_func** =  $\begin{pmatrix} v_1 \\ a^2 v_{xx_0} \end{pmatrix}$ ; начальные условия: **pinit** =  $\begin{pmatrix} x \\ x^2 \end{pmatrix}$ ; граничные условия **bc\_func** =  $\begin{pmatrix} pinit_0(0) & pinit_0(x_{end}) & "D" \\ "NA" & "NA" & "D" \end{pmatrix}$ .

3. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с ненулевыми граничными условиями на квадратной сетке размером  $(\mathbf{M} + 1) \times (\mathbf{M} + 1)$  точек ( $\mathbf{M} = 32$ ), если в точках (10, 15), (25, 10) и (5, 20) матрицы правой части уравнения заданы значения  $\mathbf{F}_{10, 15} = -7000$ ,  $\mathbf{F}_{25, 10} = 5000$  и  $\mathbf{F}_{5, 20} = -10000$ , параметры уравнения  $b_{i,k} = c_{i,k} = d_{i,k} = a_{i,k} = \sqrt{i}\sqrt{k}$ ;  $e_{i,k} = -4a_{i,k}$ , а матрица граничных условий имеет вид  $\mathbf{v}_{i,k} = -\mathbf{k} \times (1 + \mathbf{i}) / 6$ , а  $\mathbf{i}, \mathbf{k} = 0, 1, \dots, \mathbf{M}$ .

- Уравнение параболического типа  $\mathbf{u}_t = \mathbf{u}_{xx} + 3\mathbf{u}_x$  с граничными условиями Неймана  $\mathbf{u}_x(0, t) = -0.3t$ ;  $\mathbf{u}_x(\mathbf{L}, t) = 0.2t$  и начальными условиями

$u(x, 0) = \cos\left(\pi \frac{x-1}{L}\right) + \frac{1}{2} \sin\left(4\pi \frac{x+1}{L}\right)$  при  $L = 4$  и для конечной точки по времени  $T = 2$ . Использовать количество точек для решения по координате –  $N_x = 30$ , и по времени –  $N_t = 20$ .

- Одномерное волновое уравнение (использовать PDE solve block)  $v_t(x, t) = a^2 w_{xx}(x, t)$  с ограничением  $w_t(x, t) = v(x, t)$  при  $a = 1/2$  в диапазонах по координате  $[0, 1.2]$  и по времени  $[0, 3/2]$  с начальными условиями  $w(x, 0) = \cos(\pi x/L)$  и  $v(x, 0) = \text{tg}(x)$  и граничными условиями Неймана  $w_x(0, t) = \sqrt{t}$  и  $w_x(L, t) = \sin(t)$ .

4. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 32$ ), если в точках матрицы правой части уравнения заданы значения  $F_{i,j} = i^2 - j^2$ .

- Уравнение параболического типа  $u_t = u_{xx} - 2u_x$  с граничными условиями Неймана  $u_x(0, t) = 0.5t$ ;  $u_x(L, t) = 0.7t$  и начальными условиями

$u(x, 0) = \ln\left[1 + \frac{2}{3} \sin\left(4\pi \frac{x+1}{L}\right)\right]$  при  $L = 3$  и для конечной точки по времени  $T = 4$ . Использовать количество точек для решения по координате –  $N_x = 30$ , и по времени –  $N_t = 40$ .

- Система: алгебраическое уравнение  $0 = v(x, t) + ax^2t$  и одномерное волновое уравнение (использовать функцию numol)  $v_t(x, t) = a^2 v_{xx}(x, t)$  при  $a = -2^2$  в диапазонах по координате  $[x_0 = 0, x_{end} = 2.1]$  и по времени  $[t_0 = 0, t_{end} = 3.15]$ . Количество дифференциальных уравнений **num\_pde** = 2; количество алгебраических уравнений **num\_pae** = 1. Правая часть

функции **pde\_func** =  $\begin{pmatrix} v_1 \\ a^2 v_{xx_0} \\ v_1 + ax^2t \end{pmatrix}$ ; начальные условия: **pinit** =  $\begin{pmatrix} 1+x^2 \\ \sqrt{1+x} \\ 1-x^2 \end{pmatrix}$ ; граничные условия Неймана **bc\_func** =  $\begin{pmatrix} pinit_0(0) & pinit_0(x_{end}) & "N" \\ "NA" & "NA" & "D" \end{pmatrix}$ .

5. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 32$ ), если в точках матрицы правой части уравнения заданы значения  $F_{i,j} = 2 \times j - i$ .

- Уравнение параболического типа  $u_t = u_{xx} + 3u_x$  с граничными условиями Неймана  $u_x(0, t) = -5t$ ;  $u_x(L, t) = 0.7t$  и начальными условиями

$u(x, 0) = e^{1 - \frac{2}{3} \sin\left(2\pi \frac{x}{L}\right)}$  при  $L = 3$  и для конечной точки по времени  $T = 3$ . Использовать количество точек для решения по координате –  $N_x = 30$ , и по времени –  $N_t = 40$ .

- Решить систему: алгебраическое уравнение  $\mathbf{0} = \mathbf{v}(\mathbf{x}, \mathbf{t}) - \mathbf{a}\mathbf{x}\mathbf{t}$  и одномерное волновое уравнение  $\mathbf{v}_t(\mathbf{x}, \mathbf{t}) = \mathbf{a}^2 \mathbf{v}_{xx}(\mathbf{x}, \mathbf{t})$  (использовать функцию **numol**) при  $\mathbf{a} = 2^{-1}$  в диапазонах по координате  $[\mathbf{x}_0 = \mathbf{0}, \mathbf{x}_{\text{end}} = \mathbf{2}]$  и по времени  $[\mathbf{t}_0 = \mathbf{0}, \mathbf{t}_{\text{end}} = \mathbf{3}]$ . Количество дифференциальных уравнений **num\_pde** = 2; количество алгебраических уравнений **num\_pae** = 1. Правая

часть функции **pde\_func** =  $\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{a}^2 \mathbf{v}_{xx_0} \\ \mathbf{v}_1 - \mathbf{a}\mathbf{x}\mathbf{t} \end{pmatrix}$ ; начальные условия: **pinit** =  $\begin{pmatrix} 1 + \mathbf{x}^2 \\ 2\sqrt{1 + \mathbf{x}} \\ 1 - \mathbf{x}^2 \end{pmatrix}$ ;

граничные условия Дирихле **bc\_func** =  $\begin{pmatrix} \text{pinit}_0(0) & \text{pinit}_0(\mathbf{x}_{\text{end}}) & "D" \\ "NA" & "NA" & "D" \end{pmatrix}$ .

6. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с ненулевыми граничными условиями на квадратной сетке размером  $(\mathbf{M} + 1) \times (\mathbf{M} + 1)$  точек ( $\mathbf{M} = \mathbf{32}$ ), если в точках **(30, 10)**, **(5, 15)** и **(5, 30)** матрицы правой части уравнения заданы значения **F<sub>30, 10</sub> = -1200**, **F<sub>5, 15</sub> = 1500** и **F<sub>5, 30</sub> = -1300**, параметры уравнения **b<sub>i,k</sub> = c<sub>i,k</sub> = d<sub>i,k</sub> = a<sub>i,k</sub> = ln(1 + i\*k)**, а матрица граничных условий имеет вид **v<sub>i, k</sub> =  $\sqrt{i}$  × (1 - k)**, а **i, k = 0, 1, ..., M**.

- Уравнение параболического типа  $\mathbf{u}_t = \mathbf{u}_{xx} - \mathbf{u}_x / 7$  с граничными условиями Дирихле  $\mathbf{u}(0, \mathbf{t}) = \mathbf{t}^2 / 2$ ;  $\mathbf{u}(\mathbf{L}, \mathbf{t}) = \sqrt{\mathbf{t}}$  и начальными условиями  $\mathbf{u}(\mathbf{x}, 0) = \ln\left[1 - \sin\left(\frac{\mathbf{x} + 1}{\mathbf{L}}\right) + \mathbf{x}^2\right]$  при  $\mathbf{L} = \mathbf{2}$  и для конечной точки по времени **T = 3**. Использовать количество точек для решения по координате – **N<sub>x</sub> = 30**, и по времени – **N<sub>t</sub> = 60**.

- Одномерное волновое уравнение (использовать PDE solve block)  $\mathbf{v}_t(\mathbf{x}, \mathbf{t}) = \mathbf{a}^2 \mathbf{w}_{xx}(\mathbf{x}, \mathbf{t})$  с ограничением  $\mathbf{w}_t(\mathbf{x}, \mathbf{t}) = 2 \mathbf{v}(\mathbf{x}, \mathbf{t})$  при  $\mathbf{a} = 2^{-3}$  в диапазонах по координате **[0, 1.2]** и по времени **[0, 3.1]** с начальными условиями  $\mathbf{w}(\mathbf{x}, 0) = \sin(\mathbf{x} / \mathbf{L})$  и  $\mathbf{v}(\mathbf{x}, 0) = \cos(\mathbf{x})$  и граничными условиями Дирихле  $\mathbf{w}(0, \mathbf{t}) = \sqrt{\mathbf{t}}$  и  $\mathbf{w}(\mathbf{L}, \mathbf{t}) = \mathbf{t}^2$ .

7. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(\mathbf{M} + 1) \times (\mathbf{M} + 1)$  точек ( $\mathbf{M} = \mathbf{32}$ ), если в точках матрицы правой части уравнения заданы значения **F<sub>i, i</sub> = 1000 + j - i<sup>2</sup>**.

- Уравнение параболического типа  $\mathbf{u}_t = \mathbf{u}_{xx} + \mathbf{u}_x / 3$  с граничными условиями Дирихле  $\mathbf{u}(0, \mathbf{t}) = 1 / (1 + \mathbf{t})$ ;  $\mathbf{u}(\mathbf{L}, \mathbf{t}) = \mathbf{t}$  и начальными условиями  $\mathbf{u}(\mathbf{x}, 0) = \lg(x^2 - \mathbf{x} + 1)$  при  $\mathbf{L} = \mathbf{1.2}$  и для конечной точки по времени **T = 3**. Использовать количество точек для решения по координате – **N<sub>x</sub> = 30**, и по времени – **N<sub>t</sub> = 60**.

- Одномерное волновое уравнение (использовать PDE solve block)  $\mathbf{v}_t(\mathbf{x}, \mathbf{t}) = \mathbf{a}^2 \mathbf{w}_{xx}(\mathbf{x}, \mathbf{t})$  с ограничением  $\mathbf{w}_t(\mathbf{x}, \mathbf{t}) = -2\mathbf{v}(\mathbf{x}, \mathbf{t})$  при  $\mathbf{a} = 1 / 3^2$  в диа-

пазонах по координате  $[0, 1.5]$  и по времени  $[0, 1.7]$  с начальными условиями  $w(x, 0) = \cos[(1 + x) / L]$  и  $v(x, 0) = \ln(1 + x)$  и граничными условиями Неймана  $w_x(0, t) = t^2$  и  $w_x(L, t) = \ln(1 + t)$ .

8. Решить дифференциальные уравнения в частных производных:

• Уравнение Пуассона с ненулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 32$ ), если в точках  $(10, 10)$ ,  $(25, 5)$ ,  $(25, 25)$  и  $(5, 20)$  матрицы правой части уравнения заданы значения  $F_{10, 10} = -1700$ ,  $F_{25, 5} = 1900$ ,  $F_{25, 25} = -2700$  и  $F_{5, 20} = -2500$ , параметры уравнения  $b_{i,k} = c_{i,k} = d_{i,k} = a_{i,k} = i + k$ , а матрица граничных условий имеет вид  $v_{i,k} = -\sqrt{i} (1 - \sqrt{k})$ , а  $i, k = 0, 1, \dots, M$ .

• Уравнение параболического типа  $u_t = u_{xx} - 0.8u_x$  с граничными условиями Неймана  $u_x(0, t) = \sin t$ ;  $u_x(L, t) = \cos t$  и начальными условиями

$$u(x, 0) = \ln \left[ 1 - \frac{2}{3} \cos \left( 2\pi \frac{x}{L} \right) \right] \quad \text{при } L = 3 \text{ и для конечной точки по времени } T =$$

3. Использовать количество точек для решения по координате –  $N_x = 30$ , и по времени –  $N_t = 40$ .

• Система: алгебраическое уравнение  $0 = v^2(x, t) + x^2 t v(x, t)$  и одномерное волновое уравнение (использовать функцию `numol`)  $v_t(x, t) = a^2 v_{xx}(x, t)$  при  $a = 2^{-1}$  в диапазонах по координате  $[x_0 = 0, x_{\text{end}} = 4]$  и по времени  $[t_0 = 0, t_{\text{end}} = 5]$ . Количество дифференциальных уравнений `num_pde` = 2; количество алгебраических уравнений `num_pae` = 1. Правая

часть функции `pde_func` = 
$$\begin{pmatrix} v_1 \\ a^2 v_{xx_0} \\ v_1^2 + x^2 t v_1 \end{pmatrix}; \text{ начальные условия: } \mathbf{pinit} = \begin{pmatrix} \sin(x^2) \\ \sqrt{x+3} \\ -x^2 \end{pmatrix};$$

граничные условия Неймана `bc_func` = 
$$\begin{pmatrix} \mathbf{pinit}_0(0) & \mathbf{pinit}_0(x_{\text{end}}) & "N" \\ "NA" & "NA" & "D" \end{pmatrix}.$$

9. Решить дифференциальные уравнения в частных производных:

• Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 32$ ), если в точках матрицы правой части уравнения заданы значения  $F_{i,j} = j \times i - 25 \sqrt{i+j}$ .

• Уравнение параболического типа  $u_t = u_{xx} + 3 u_x / 7$  с граничными условиями Дирихле  $u(0, t) = 0.215 - t$ ;  $u(L, t) = t - 0.144$  и начальными условиями  $u(x, 0) = 1 - \arctg(1 - x + 2x^2 - 0.3x^3)$  при  $L = 6$  и для конечной точки по времени  $T = 3$ . Использовать количество точек для решения по координате –  $N_x = 60$ , и по времени –  $N_t = 60$ .

• Одномерное волновое уравнение (использовать `PDE solve block`)  $v_t(x, t) = a^2 w_{xx}(x, t)$  с ограничением  $w_t(x, t) = -0.2 v(x, t)$  при  $a = 3^{-2}$  в диапазонах по координате  $[0, 1.7]$  и по времени  $[0, 1.2]$  с начальными условиями

ями  $w(x, 0) = 1 - \sin(x)$  и  $v(x, 0) = x^2 - 1$  и граничными условиями Дирихле  $w(0, t) = t^2 + 1$  и  $w(L, t) = 0$ .

10. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с ненулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 48$ ), если в точках  $(01, 10)$ ,  $(40, 5)$ ,  $(35, 35)$  и  $(5, 40)$  матрицы правой части уравнения заданы значения  $F_{05, 10} = 6$ ,  $F_{40, 5} = 7$ ,  $F_{35, 35} = -10$  и  $F_{5, 40} = -2$ , параметры уравнения  $b_{i,k} = c_{i,k} = d_{i,k} = a_{i,k} = i / (k + 1)$ , а матрица граничных условий имеет вид  $v_{i,k} = -e^{-i} (1 + \sqrt[3]{k})$ , а  $i, k = 0, 1, \dots, M$ .

- Уравнение параболического типа  $u_t = u_{xx} + 1/2 u_x$  с граничными условиями Неймана  $u_x(0, t) = -2\sqrt{x}$ ;  $u_x(L, t) = 1 - 0.27t^2$  и начальными условиями  $u(x, 0) = (x - 1)^2$  при  $L = 3.1$  и для конечной точки по времени  $T = 2.2$ . Использовать количество точек для решения по координате –  $N_x = 30$ , и по времени –  $N_t = 40$ .

- Одномерное волновое уравнение (использовать PDE solve block)  $v_t(x, t) = a^2 w_{xx}(x, t)$  с ограничением  $w_t(x, t) = -0.2v(x, t)$  при  $a = 1.09^2$  в диапазонах по координате  $[0, 1.5]$  и по времени  $[0, 1.7]$  с начальными условиями  $w(x, 0) = \ln[(1 + x) / L]$  и  $v(x, 0) = \sin[(1 + x) / L]$  и граничными условиями Неймана  $w_x(0, t) = \sqrt{t}$  и  $w_x(L, t) = \arctg(1 + t)$ .

11. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 32$ ), если в точках матрицы правой части уравнения заданы значения  $F_{i,j} = j / (i + 1)$ ;  $i = 0 \dots M - 1$ ;  $j = 0 \dots M - 1$ , и  $F_{15, 15} = 50$ .

- Уравнение параболического типа  $u_t = u_{xx} - 0.75 u_x$  с граничными условиями Неймана  $u_x(0, t) = 1 - 2\sqrt{x}$ ;  $u_x(L, t) = 2.008 - \sqrt[5]{t}$  и начальными условиями  $u(x, 0) = \sqrt[3]{x+1}$  при  $L = 7.1$  и для конечной точки по времени  $T = 42$ . Использовать количество точек для решения по координате –  $N_x = 71$ , и по времени –  $N_t = 42$ .

- Система: алгебраическое уравнение  $0 = (x + 1) v^2(x, t) + t v(x, t)$  и одномерное волновое уравнение (использовать функцию numol)  $v_t(x, t) = a^2 v_{xx}(x, t)$  при  $a = \sqrt{3}$  в диапазонах по координате  $[x_0 = 0, x_{end} = 4]$  и по времени  $[t_0 = 0, t_{end} = 3]$ . Количество дифференциальных уравнений num\_pde = 2; количество алгебраических уравнений num\_pae = 1. Правая

часть функции pde\_func = 
$$\begin{pmatrix} v_1 \\ a^2 v_{xx_0} \\ (x+1)v_1^2 + tv_1 \end{pmatrix};$$
 начальные условия: pinit = 
$$\begin{pmatrix} \sqrt{x} \\ x \\ 0 \end{pmatrix}$$

; граничные условия Дирихле bc\_func = 
$$\begin{pmatrix} pinit_0(0) & pinit_0(x_{end}) & "D" \\ "NA" & "NA" & "D" \end{pmatrix}.$$

12. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с ненулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 48$ ), если в точках (15, 20), (10, 40), (24, 24) и (40, 40) матрицы правой части уравнения заданы значения  $F_{15, 20} = -6$ ,  $F_{10, 40} = 9$ ,  $F_{24, 24} = -11$  и  $F_{40, 40} = 5$ , параметры уравнения  $b_{i,k} = c_{i,k} = d_{i,k} = a_{i,k} = (k + 15) / (i + 5)$ , а матрица граничных условий имеет вид  $v_{i,k} = -e^{k-i} (1 + \ln(1 + i + k))$ , а  $i, k = 0, 1, \dots, M$ .

- Уравнение параболического типа  $u_t = u_{xx} + 2 u_x / 5$  с граничными условиями Дирихле  $u(0, t) = 0.54 - t$ ;  $u(L, t) = t + 0.99$  и начальными условиями  $u(x, 0) = \cos(1 / (1 + x))$  при  $L = 6$  и для конечной точки по времени  $T = 3$ . Использовать количество точек для решения по координате –  $N_x = 60$ , и по времени –  $N_t = 60$ .

- Система: алгебраическое уравнение  $0 = t^2 v(x, t) + x t \sqrt{v(x, t)}$  и одномерное волновое уравнение (использовать функцию **numol**)  $v_t(x, t) = a^2 v_{xx}(x, t)$  при  $a = 3 / 5$  в диапазонах по координате  $[x_0 = 0, x_{end} = 2]$  и по времени  $[t_0 = 0, t_{end} = 1.5]$ . Количество дифференциальных уравнений **num\_pde** = 2; количество алгебраических уравнений **num\_pae** = 1.

Правая часть функции **pde\_func** = 
$$\begin{pmatrix} v_1 \\ a^2 v_{xx_0} \\ t^2 v_1 + x t \sqrt{v_1} \end{pmatrix};$$
 начальные условия:

**pinit** = 
$$\begin{pmatrix} \sqrt{\cos(x^2)} \\ e^{-x} \\ x \end{pmatrix};$$
 граничные условия Неймана **bc\_func** = 
$$\begin{pmatrix} pinit_0(0) & pinit_0(x_{end}) & "N" \\ "NA" & "NA" & "D" \end{pmatrix}.$$

13. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 64$ ), если в точ-

ках матрицы правой части уравнения заданы значения  $F_{i,i} = \frac{\sin\left(\frac{5j}{M+1}\right)}{\cos(i)+1}$ .

- Уравнение параболического типа  $u_t = u_{xx} - 1 u_x / 7$  с граничными условиями Дирихле  $u(0, t) = -0.785 - t$ ;  $u(L, t) = t + 0.59$  и начальными условиями  $u(x, 0) = \arctg(1 / (x^2 + 7x - 1))$  при  $L = 2$  и для конечной точки по времени  $T = 3$ . Использовать количество точек для решения по координате –  $N_x = 60$ , и по времени –  $N_t = 60$ .

- Одномерное волновое уравнение (использовать PDE solve block)  $v_t(x, t) = a^2 w_{xx}(x, t)$  с ограничением  $w_t(x, t) = 0.5 v(x, t)$  при  $a = 10^{-1}$  в диа-

пазонах по координате  $[0, 3.7]$  и по времени  $[0, 1.2]$  с начальными условиями  $\mathbf{w}(\mathbf{x}, 0) = \ln(\mathbf{x}+1.1)$  и  $\mathbf{v}(\mathbf{x}, 0) = \sqrt{x+2}$  и граничными условиями Дирихле  $\mathbf{w}(0, t) = t^2 + 2t + 0.095$  и  $\mathbf{w}(L, t) = 1.569 + \sqrt{t}$ .

14. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с ненулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 32$ ), если в точках  $(20, 20)$ ,  $(10, 30)$ ,  $(25, 25)$  и  $(30, 10)$  матрицы правой части уравнения заданы значения  $F_{20, 20} = 36$ ,  $F_{10, 30} = 19$ ,  $F_{25, 25} = 101$  и  $F_{30, 10} = 105$ , параметры уравнения  $\mathbf{b}_{i,k} = \mathbf{c}_{i,k} = \mathbf{d}_{i,k} = \mathbf{a}_{i,k} = (\mathbf{k} + 15) \mathbf{i}$ , а матрица граничных

условий имеет вид  $\mathbf{v}_{i,k} = \frac{\exp\left(\cos\left(\sqrt{|k-i|}\right)\right)}{\exp\left(\cos\left(\sqrt{|k-i|}\right)\right)}$ , а  $\mathbf{i}, \mathbf{k} = 0, 1, \dots, M$ .

- Уравнение параболического типа  $\mathbf{u}_t = \mathbf{u}_{xx} + 0.3 \mathbf{u}_x$  с граничными условиями Неймана  $\mathbf{u}_x(0, t) = 1 - 2\sin(\sqrt{x})$ ;  $\mathbf{u}_x(L, t) = 1 - \arctg(t)$  и начальными условиями  $\mathbf{u}(\mathbf{x}, 0) = \sqrt[3]{|x-3|}$  при  $L = 17$  и для конечной точки по времени  $T = 3$ . Использовать количество точек для решения по координате –  $N_x = 70$  и по времени –  $N_t = 40$ .

- Система: алгебраическое уравнение  $0 = (t + 1) \mathbf{v}^2(\mathbf{x}, t) - \mathbf{a} \mathbf{x} \mathbf{v}(\mathbf{x}, t)$  и одномерное волновое уравнение (использовать функцию **numol**)  $\mathbf{v}_t(\mathbf{x}, t) = \mathbf{a}^2 \mathbf{v}_{xx}(\mathbf{x}, t)$  при  $\mathbf{a} = \sqrt[3]{3}$  в диапазонах по координате  $[\mathbf{x}_0 = 0, \mathbf{x}_{end} = 4]$  и по времени  $[\mathbf{t}_0 = 0, \mathbf{t}_{end} = 3]$ . Количество дифференциальных уравнений **num\_pde** = 2; количество алгебраических уравнений **num\_pae** = 1. Правая

часть функции **pde\_func** =  $\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{a}^2 \mathbf{v}_{xx_0} \\ ((t+1)\mathbf{v}_1^2 + \mathbf{a}\mathbf{x}\mathbf{v}_1) \end{pmatrix}$ ; начальные условия:

**pinit** =  $\begin{pmatrix} \sin(x^2) \\ 1 - \cos(x) \\ \arctg(x+3) \end{pmatrix}$ ; граничные условия Дирихле

**bc\_func** =  $\begin{pmatrix} pinit_0(0) & pinit_0(x_{end}) & "D" \\ "NA" & "NA" & "D" \end{pmatrix}$ .

15. Решить дифференциальные уравнения в частных производных:

- Уравнение Пуассона с нулевыми граничными условиями на квадратной сетке размером  $(M + 1) \times (M + 1)$  точек ( $M = 64$ ), если в точках матрицы правой части уравнения заданы значения  $\mathbf{F}_{i,i} = \frac{\sin\left(\frac{3i+2j}{M+1}\right)}{\cos(j) + \sin(i)}$ .



• Уравнение параболического типа  $u_t = u_{xx} + 2 u_x / 9$  с граничными условиями Дирихле  $u(0, t) = 0.54 - t$ ;  $u(L, t) = t + 0.907$  и начальными условиями  $u(x, 0) = \exp(\cos(7x - 1))$  при  $L = 2$  и для конечной точки по времени  $T = 3$ . Использовать количество точек для решения по координате –  $N_x = 60$ , и по времени –  $N_t = 60$ .

Одномерное волновое уравнение (использовать PDE solve block)  $v_t(x, t) = a^2 w_{xx}(x, t)$  с ограничением  $w_t(x, t) = -1.2v(x, t)$  при  $a = 0.137$  в диапазонах по координате  $[0, 3]$  и по времени  $[0, 2]$  с начальными условиями

$w(x, 0) = \ln\left(\frac{\sqrt{x+2}}{L}\right)$  ( $L = 3$ ) и  $v(x, 0) = \cos\left(\frac{1+x+x^2}{L}\right)$  и граничными условиями Неймана  $w_x(0, t) = -0.752 + t^2$  и  $w_x(L, t) = \ln(1+t) - 0.294$ .

### 2.3 Контрольные вопросы

1. С помощью какой функции можно решить задачу Коши в MathCAD?

2. Можно ли использовать смешанные начальные или граничные условия при решении обыкновенного дифференциального уравнения с помощью функции Odesolve?

3. Какую функцию следует использовать для численного решения краевой задачи в MathCAD?

4. Перечислите, какие функции можно использовать для численного решения задачи Коши для дифференциальных уравнений и систем.

5. Укажите назначение аргументов функций, которые можно использовать для численного решения задачи Коши для дифференциальных уравнений и систем.

6. Перечислите функции, которые используются для решения дифференциального уравнения n-го порядка (краевая задача).

7. Опишите метод численного решения двухточечной краевой задачи (дифференциального уравнения n-го порядка) с условиями на краях интервала.

8. Опишите метод численного решения двухточечной краевой задачи (дифференциального уравнения n-го порядка) с условиями внутри интервала.

9. Опишите метод численного решения двумерного уравнения Пуассона с нулевыми граничными условиями.

10. Опишите метод численного решения двумерного уравнения Пуассона с ненулевыми граничными условиями.

11. Опишите метод численного решения уравнения параболического типа с граничными условиями Дирихле или Неймана.

12. Опишите метод численного решения одномерного волнового уравнения с использованием PDE solve block'a.

13. Укажите назначение аргументов функции `Pdesolve` при решении одномерного волнового уравнения.

14. Опишите метод численного решения одномерного волнового уравнения с использованием функции `numol`.

15. Укажите назначение аргументов функции `numol` при решении одномерного волнового уравнения.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. ЛАБОРАТОРНАЯ РОБОТА 1	
Программирование и регрессия в MathCAD .....	4
1.1. Общие сведения.....	4
1.1.1. Программирование на MathCAD «без программирования».....	4
1.1.2. Язык программирования MathCAD.....	5
1.1.3. Примеры программирования .....	17
1.1.4. Функции для проведения регрессии .....	22
1.2. Порядок выполнения работы.....	27
1.2.1. Задания для выполнения работы.....	27
1.2.2. Варианты заданий .....	28
1.3. Контрольные вопросы.....	45
2. ЛАБОРАТОРНАЯ РОБОТА 2	
Решение дифференциальных уравнений и систем .....	46
2.1. Общие сведения.....	46
2.1.1. Решение дифференциального уравнения.....	46
2.1.2. Численное решение задачи Коши для дифференциальных уравнений и систем.....	47
2.1.3. Решение краевых задач для обыкновенных дифференциальных уравнений.....	50
2.1.4. Уравнения в частных производных.....	53
2.2. Порядок выполнения работы.....	62
2.2.1. Задания для выполнения работы.....	62
2.2.2. Варианты заданий.....	62
2.3. Контрольные вопросы.....	72

Учебное издание

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**к лабораторным работам**  
**«Программирование и решение дифференциальных уравнений**  
**в системе MathCAD»**  
**из раздела «Использование программ в среде Windows»**  
**дисциплины «Основы программирования**  
**и информационных технологий»**  
**для студентов направления подготовки**  
**6.050801 «Микро- и нанoeлектроника»**

Составители: ЗАЙЦЕВ Роман Валентинович  
КЛЕПИКОВА Екатерина Сергеевна  
МОМОТЕНКО Александра Витальевна

Ответственный за выпуск М.В. Кириченко

План 2013 р.

Підписано до друку 15.01.14. Формат 60×84 1/16. Папір друк. №2.  
Друк – ризографія. Гарнітура Times New Roman. Ум. друк. арк. 4,4.  
Обл.-вид. 5,0. Тираж 50 прим. Зам. № \_\_\_\_\_. Ціна договірна.

---

Видавничий центр НТУ “ХПІ”. 61002, Харків, вул. Фрунзе, 21.  
Свідоцтво про державну реєстрацію ДК № 116 від 10.07.2000 р.

---

Друкарня НТУ “ХПІ”. 61002, Харків, вул. Фрунзе, 21.